

ANALYSIS OF ROAD TRAFFIC FATAL ACCIDENT USING DATA MINING TECHNIQUES

Submitted in partial fulfillment of the requirements
for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

Kummara Naresh (38110277)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119**

MARCH – 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY



(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Kummara Naresh (38110277)** who carried out the project entitled “**Analysis of Road Traffic Accident using Data Mining Techniques**” under my supervision from November 2021 to March 2022.

Internal Guide

Mr. Dr. A. Pravin, M.E, ph.d

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **Kummara Naresh (Reg No:38110277)** hereby declare that the Project Report

entitled “**Analysis of Road Traffic Accident using Data Mining Techniques**” done by us under the guidance of **Mr. Dr. A. Pravin ,M.E ,Ph.d**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D., Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D.** , Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Mr. Dr. A. Pravin, M.E, Ph.d**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This report presents the results from the research study on applying large scale data mining methods into analysis of traffic accidents on the Finnish roads. The data sets collected from traffic fatal accidents are huge, multidimensional, and heterogeneous. Moreover, they may contain incomplete and erroneous values, which make its exploration and understanding a very demanding task. The target data of this study was collected by the Finnish Road Administration Datasets. The intention is to investigate the usability of robust clustering, association and frequent itemsets, and visualization methods to the road traffic accident analysis. While the results show that the selected data mining methods are able to produce understandable patterns from the data, finding more fertilized information could be enhanced with more detailed and comprehensive data sets. *K-means* algorithm takes accident frequency count as a parameter to cluster the locations. Then we used association rule mining to characterize these Surface Condition. The rules revealed different factors associated with road accidents at different drunk and drive with varying accident frequencies. The association rules for high-frequency accident location disclosed that intersections on highways are more dangerous for every type of fatal accidents.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 OVERVIEW	1
2	LITERATURE SURVEY	3
	2.1. RELATED WORK	3
3	AIM AND SCOPE OF THE PROJECT	5
	3.1. EXISTING SYSTEM	5
	3.2. PROPOSED SYSTEM	5
	3.3. OBJECTIVE	5
4	METHODOLOGY	7
	4.1. HARDWARE REQUIREMENTS	7
	4.2. SOFTWARE REQUIRMENTS	7
	4.3. SYSTEM ARCHITECTURE	8
	4.4. APPLICATION OF JAVA & SQL	14
	4.5. DEEP LEARNING	23
	4.6. SYSTEM DESIGN AND RESTING PLAN	24
	4.7. ALGORITHM	30
	4.8. MODULES	31
5	RESULTS AND DISCUSSION	34
	5.1. PERFORMANCE ANALYSIS	34

6	CONCLUSION	35
	6.1. CONCLUSION	35
	REFERENCES	36
	APPENDICES	38
	A. SOURCE CODE	38
	B. SCREENSHOTS	121
	C. PLAGIARISM REPORT	139

LIST OF FIGURES

Fig No.	Figure Name	Page No.
1.	SYSTEM ARCHITECURE	9
2.	ED DIAGRAM	15
3.	USE CASE DIAGRAM	18
4.	SEQUENCE DIAGRAM	20
5.	COLLABRATION DIAGRAM	21
6.	PLATFORM INDEPENDENT	22
7.	COLLECTION FRAMEWORK	23
8.	MANAGEMENT STUDIO	24
9.	QUERY EDITOR	27
10.	NEW DATABASE	28
11.	ADMIN LOGIN PAGE	32
12.	REPORT ACCIDENT MODULE	33
13.	Gantt Chart	34

LIST OF ABBREVIATIONS

ABBREVIATIONS**EXPANSION**

ANN

Artificial Neural network

BGR

Blue Green Red

RGB

Decision Tree

CNN

Convolutional neural network

MLP

Multi Layer Perceptron

VGG

Very Deep Convolutional
Networks for Large-Scale Image
Recognition

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

There are a lot of vehicles driving on the roadway every day, and traffic accidents could happen at any time anywhere. Some accident involves fatality, means people die in that accident. As human being, we all want to avoid accident and stay safe. To find out how to drive safer, data mining technique could be applied on the traffic accident dataset to find out some valuable information, thus give driving suggestion. Data mining uses many different techniques and algorithms to discover the relationship in large amount of data. It is considered one of the most important tool in information technology in the previous decades. Association rule mining algorithm is a popular methodology to identify the significant relations between the data stored in large database and also plays a very important role in frequent itemset mining. A classical association rule mining method is the Apriori algorithm who main task is to find frequent itemsets, which is the method we use to analyze the roadway traffic data. Classification in data mining methodology aims at constructing a model (classifier) from a training data set that can be used to classify records of unknown class labels. The Naive Bayes technique is one of the very basic probability-based methods for classification that is based on the Bayes' hypothesis with the presumption of independence between each pair of variables. We used the FARS dataset for our study. The Fatal Accidents Dataset contains all fatal accidents on public roads in 2007 reported to the National Highway Transportation Safety Administration. The dataset is downloaded from California Polytechnic State University and all data originally came from FARS. The dataset contains 37,248 records and 55 attributes. The data description can be found in the document FARS.

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which

operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

CHAPTER 2

LITERATURE SURVEY

2.1 RELATED WORK

Jayasudha [4] dissected auto collisions utilizing information mining strategies that could diminish the quantity of passings. The utilization of street wellbeing offices assists with diminishing the quantity of nearby and public street security executions. These reasoning clarifies a car crash in light of the idea of the street, the individual in question, and other data that might be useful in settling the case, assembling extra proof, settling on a choice, and quieting down. Worldwide Road and Accident Information Center, GLOBESAFE, ARC network site, most ideal way to gather mishap data. Set up a photograph card utilizing the site data.

Eric [7] concentrated on the impacts of street mishaps in the territory of Washington. As per one review, the loss of life in the state has expanded from 55 kilometers each hour to 65 kilometers each hour starting around 1974, an increment of 27% from 10% in states that poor person sped up. It is trusted that the impacts of huge scope variety in metropolitan and rustic regions. Mishaps have expanded in rustic regions starting around 1987, yet in metropolitan regions the quantity of mishaps at metropolitan intersections has served as in provincial regions. Mishaps are brought about by the locale (metropolitan/rustic) and the sort of (street, street) [7]. A crisis reaction program has been sent off to diminish the quantity of fatalities brought about by mishaps. Lessening the span of crisis clinical consideration casualties can save lives. The circumstance of a mishap notice and the partition of a crisis EMS notice are significant. Wounds and sicknesses are brought about by the way that they can be restored when the casualties are dealt with. The injuries can be recuperated and treated rapidly. The initial 10 minutes are known as the brilliant hour. It was 12.3 minutes for the dead and 8.4 minutes for the survivors. The loss of life was high in rustic regions as there was no quick EMS reaction here. There are numerous different variables that add to a ridiculous degree, like driving, drinking, lessening the age of the driver, the hour of the mishap, and the pay of the populace [3].

Solomon and others. al. [8] laid out different techniques for gathering mishap data, putting it on a common server, and deciphering mishaps. Data can be gathered in an assortment of business sectors, and countless business sectors, with great outcomes. This is on the grounds that data can be produced, and few can be erased, subsequently assisting with erasing data. You can indicate various qualities for a particular position, for example, association type, kind of impact, area, month, season of day, sort of vehicle, and so on, and afterward perceive how those boundaries change on schedule and how they work. In light of these qualities, the kind of mishap can be carried out. Utilizing the API Map can make the framework change and quit adulterating cycles [8].

Halfway and fractional covering by Kumar [6] to consolidate comparable mishaps. In view of the attributes of most carried out information, the K-model calculation was utilized. To see the connection between various conditions, mining laws have been set up. To start with, the data is classified into 6 segments, every one of which is intended to foresee the construction. Of the different laws that have been instituted, the ones that appeared to be the most fascinating were considered in view of how much help and trust. Studies have shown that mishaps are neighborhood, and that most mishaps happen in thickly populated regions like business sectors, clinics, and nearby settlements. The kind of vehicle meant that the idea of the mishap; By mishap, two wheels can be associated with a mishap, with at least two individuals. Visual deficiency was the principle factor in the accident [6].

Krishnaveni and Hemalata [5] utilized a similar strategy to distinguish the classifications of street mishaps. . The end-product shows that memory isn't one of a kind to different calculations [5]. Amira, Pareek, Araar. Following the starter information, the standards of the Apriori and Predictive Apriori calculations were utilized to compute the connection between the quantity of mishaps recorded in Dubai and the seriousness of the mishap. Utilizing two calculations, two phases of sequencing rules have been created and ordered to acquire fascinating standards utilizing specialized procedures. The genuine outcomes show that the accumulation rules set by the Apriori calculation are obviously superior to those set by the Predictive Apriori calculation. Utilizing the deduced calculation [2] analyzed the connection among mishaps and the seriousness of mishaps.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

3.1 EXISTING SYSTEM

The traffic accident using data mining technique that could possibly reduce the fatality rate. Using a road safety database enables to reduce the fatality by implementing road safety programs at local and national levels. Classification models to predict the severity of injury that occurred during traffic accidents. Association rules mining algorithm on a dataset about traffic accidents which was gathered from Government Traffic Office, Apriori and Predictive Apriori association rules algorithms were applied to the dataset to investigate the connection between recorded accidents and factors to accident severity.

3.2 PROPOSED SYSTEM

This paper presents our research to model the severity of injury resulting from traffic accidents using artificial neural networks and decision trees. We have applied them to an actual data set obtained from the National Automotive Sampling System (NASS) General Estimates System (GES). Experiment results reveal that in all the cases the decision tree outperforms the neural network. Our research analysis also shows that the three most important factors in fatal injury are: driver's seat belt usage, light condition of the roadway, and driver's alcohol usage. Our experiments also showed that the model for fatal and non-fatal injury performed better than other classes. The ability of predicting fatal and non-fatal injury is very important since drivers' fatality has the highest cost to society economically and socially.

3.3 OBJECTIVE

The purpose of this study was to conduct an indepth study focusing on the application of event analysis through crash investigation and reconstruction. The objectives of this study were the followings: To identify the contributory factors based on the findings obtained from crash investigation and reconstruction by using a case study. To apply an event analysis in establishing the links between the events to describe the crash scenario based on the available information.

CHAPTER 4

METHODOLOGY

4.1. HARDWARE REQUIREMENTS

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 512MB

4.2. SOFTWARE REQUIREMENTS

- Operating System - Windows XP
- Coding language - Java
- IDE - Net beans
- Database -MYSQL

4.3. SYSTEM ARCHITECTURE

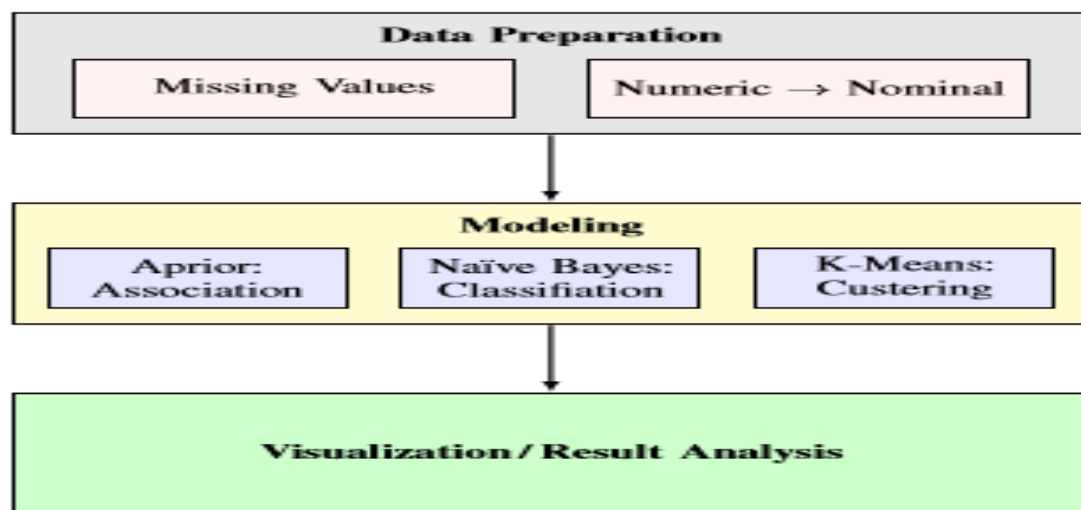


Fig. 1. System architecture

Data Flow Diagram (DFD) is a two-dimensional diagram that describes how data is processed and transmitted in a system. The graphical depiction recognizes each source of data and how it interacts with other data sources to reach a mutual output. In order to draft a data flow diagram one must

- Identify external inputs and outputs
- Determine how the inputs and outputs relate to each other
- Explain with graphics how these connections relate and what they result in.

Role of DFD:

- It is a documentation support which is understood by both programmers and nonprogrammers. As DFD postulates only what processes are accomplished not how they are performed.
- A physical DFD postulates where the data flows and who processes the data.
- It permits analyst to isolate areas of interest in the organization and study them by examining the data that enter the process and viewing how they are altered when they leave.

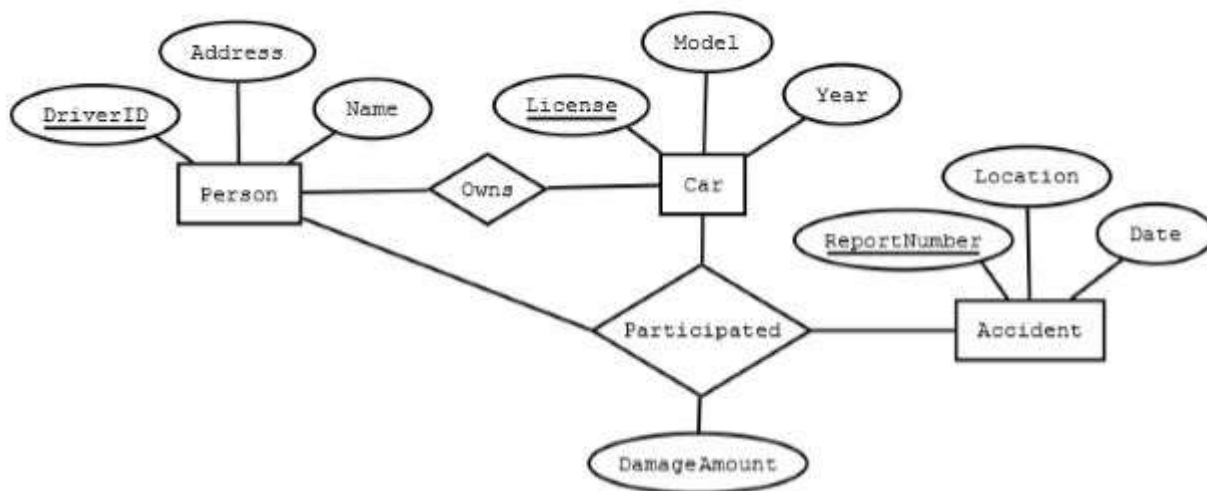


Fig.2. ER diagram

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

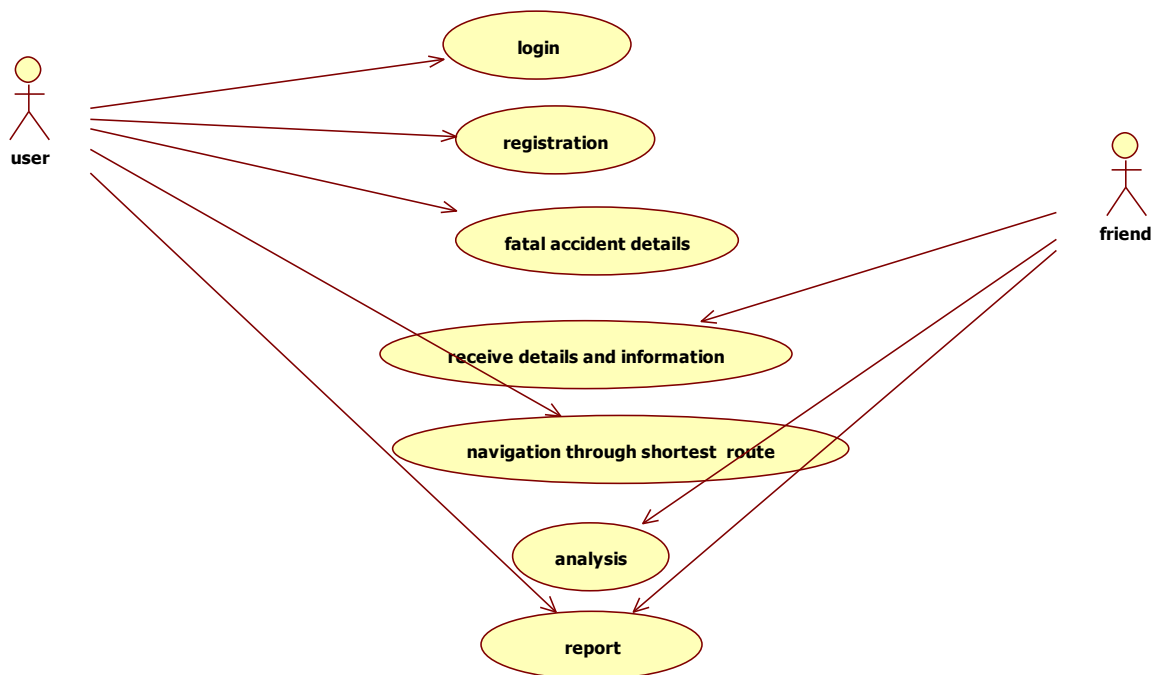


Fig.3. Use case diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

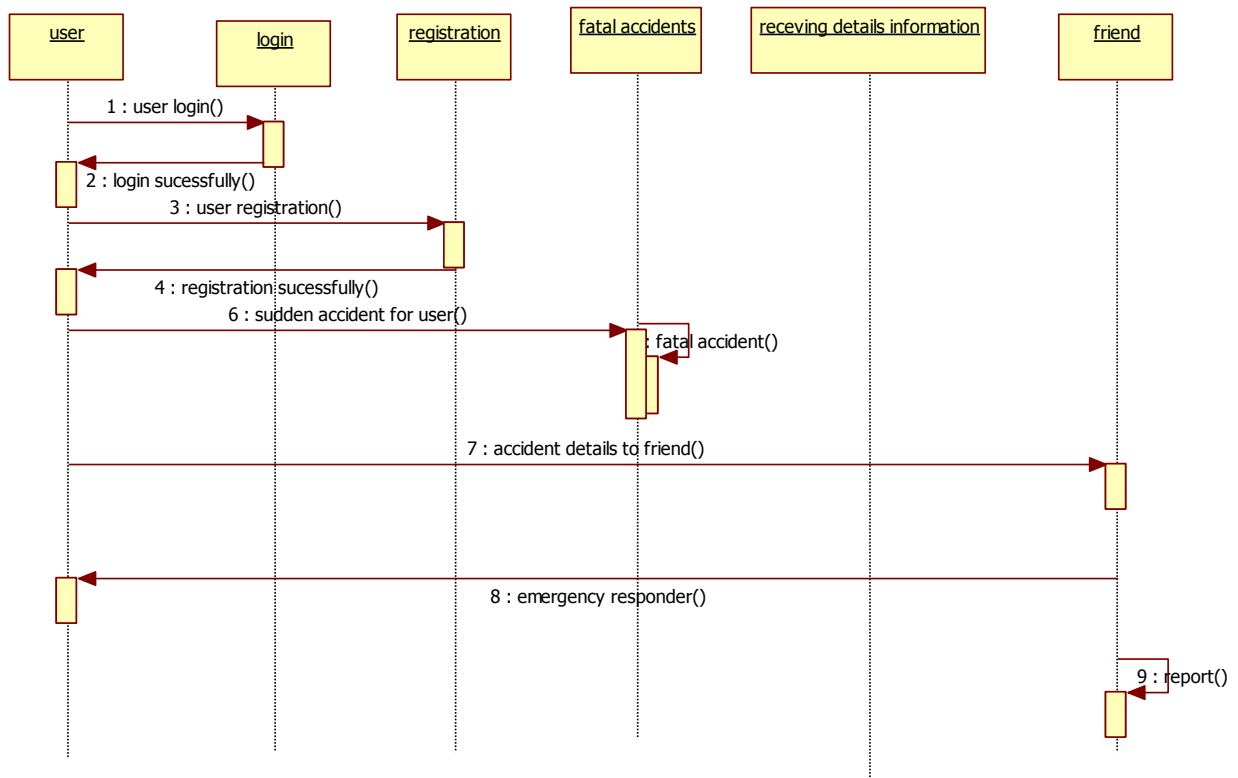


Fig.4. Sequence diagram

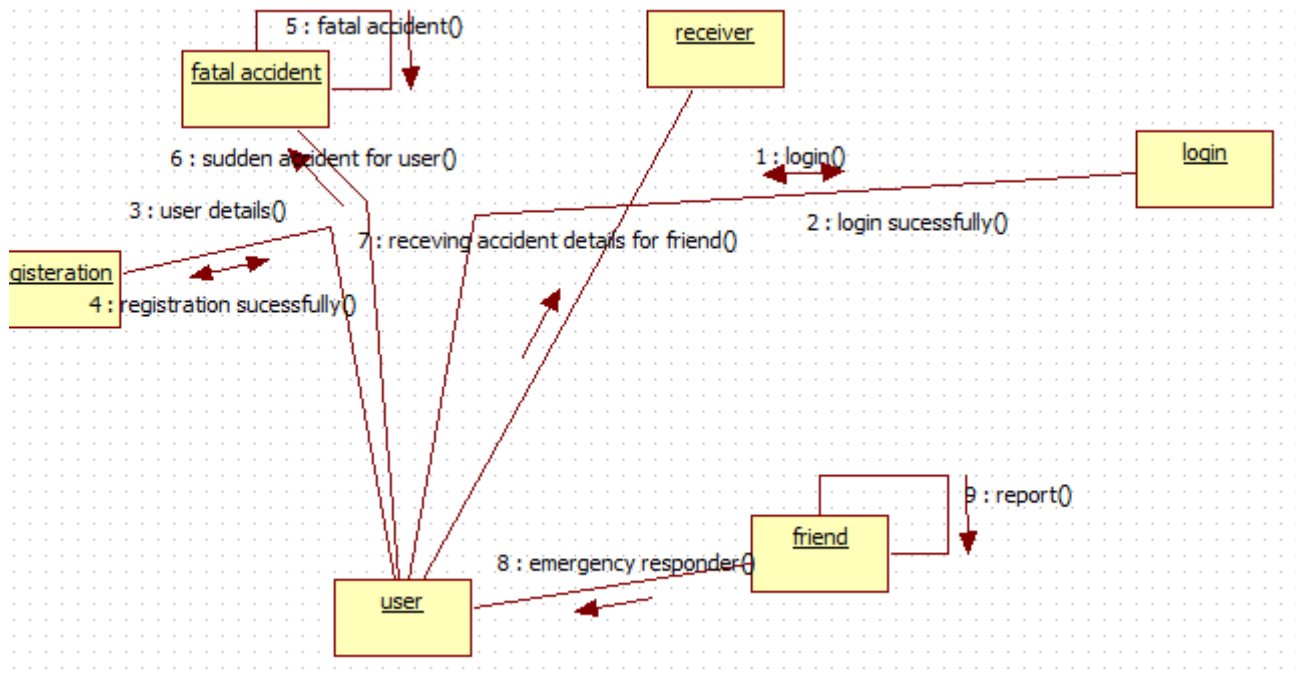


Fig.5. Collaboration diagram

4.4 APPLICATION OF JAVA & SQL

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like LinkedIn.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems

5. Robotics and games etc.

4.4.1 FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

3) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime Environment can run Java Programs.

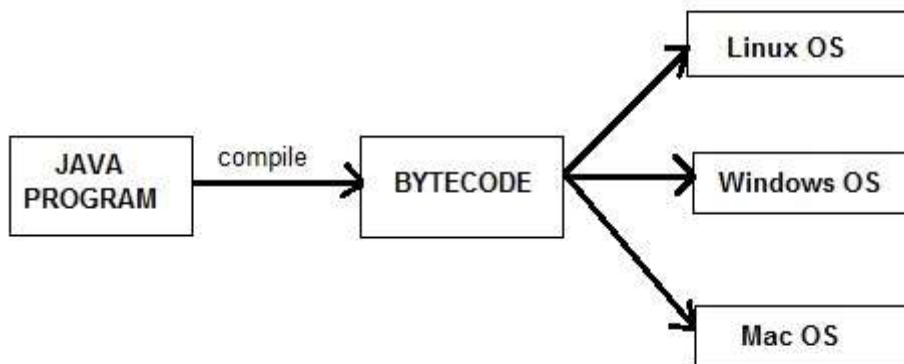


Fig.6.Platform Independent

5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7) Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

Portable

Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

COLLECTION FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.

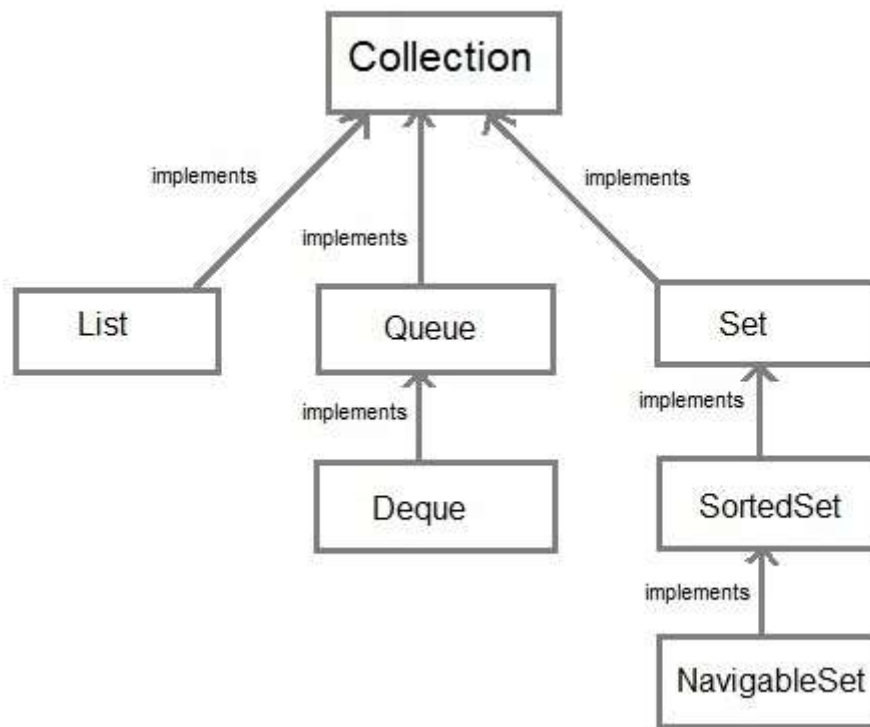


Fig.7. Collection Framework

MYSQL

MySQL, officially, but also called "My Sequel" is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks).

LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Freesoftware-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr and YouTube. Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use. SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. TSQL is very similar to standard SQL, but in addition it supports some extra functionality, built in functions, etc.

T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, date processing, mathematics, etc. SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).

SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

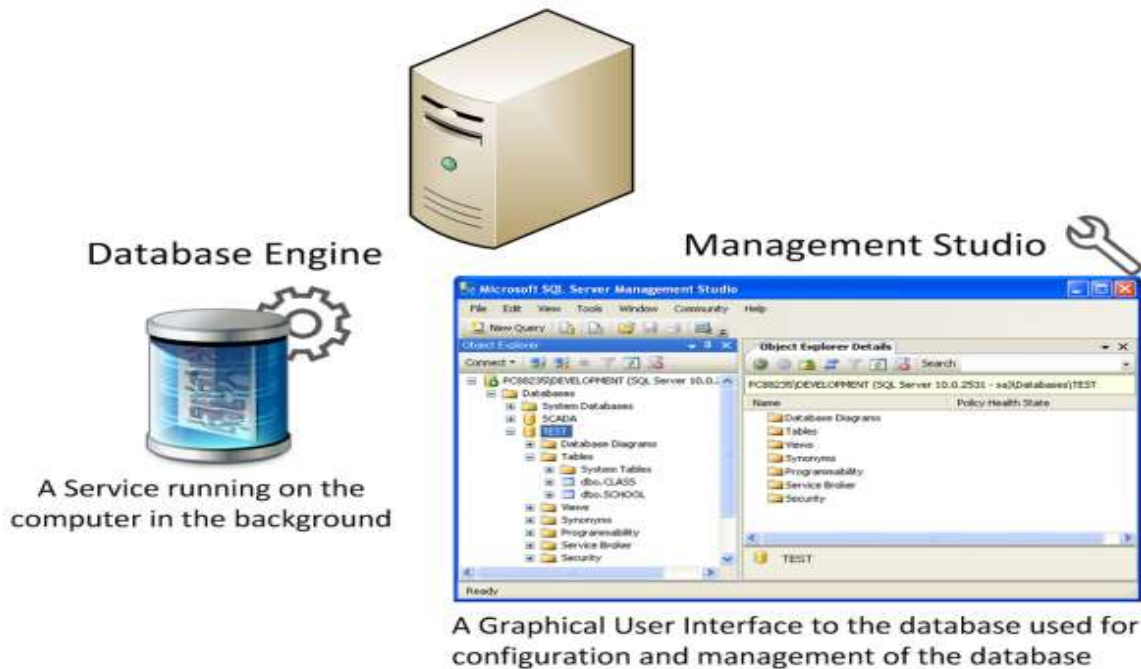


Fig.8. Management Studio

4.4. 2 SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier, version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express.

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to

create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

When creating SQL commands and queries, the “Query Editor” (select “New Query” from the Toolbar) is used (shown in the figure above). With SQL and the “Query Editor” we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).

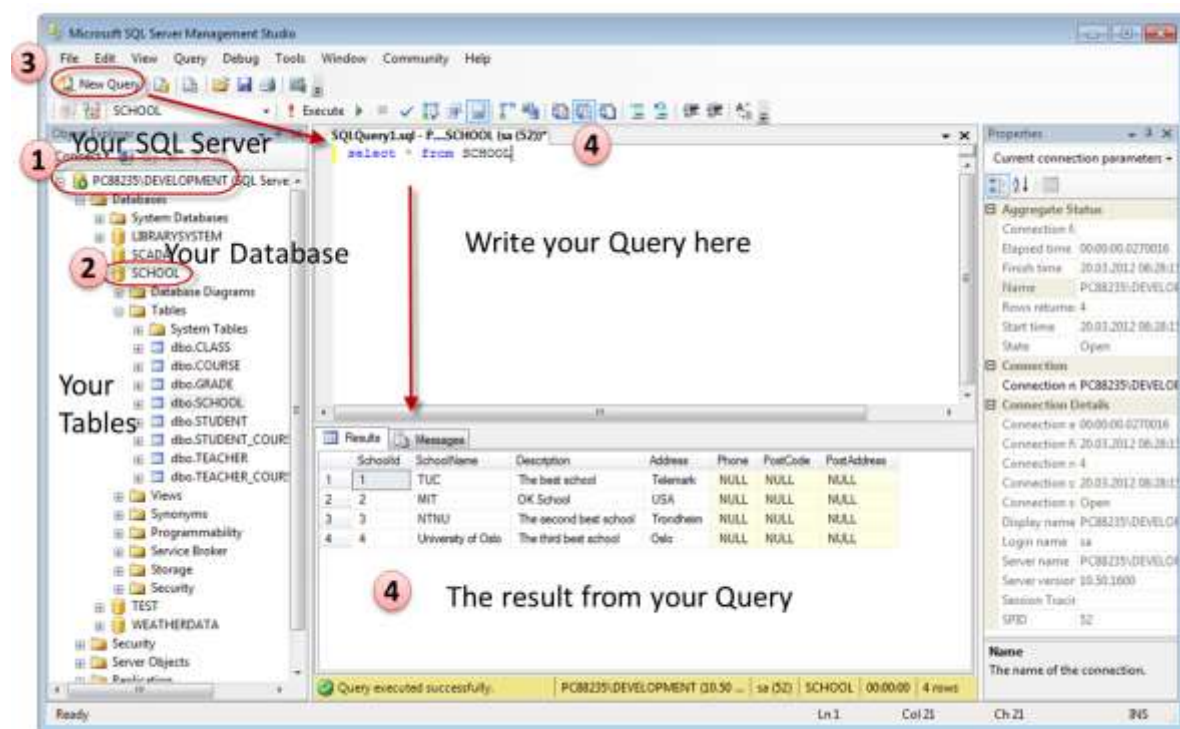


Fig.9. Query Editor

Create a new Database

It is quite simple to create a new database in Microsoft SQL Server. Just right-click on the “Databases” node and select “New Database...”



Fig.10. New database

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects,

They are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

Table

A database is a collection of data about a specific topic.

Views of Table

We can work with a table in two types,

1. Design View
2. Datasheet View

Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

Query

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

4.5 DEEP LEARNING

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.The goal of

the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

4.6 SYSTEM DESIGN AND TESTING PLAN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing
- Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing..

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging

or imposing other modules to produce unexpected results. It falls under the class of black box testing.

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the

application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

4.7 ALGORITHM

K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems. It follows a simple procedure of classifying a given data set into a number of clusters, defined by the letter "k," which is fixed beforehand. The clusters are then positioned as points and all observations or data points are associated with the nearest cluster, computed, adjusted and then the process starts over using the new adjustments until a desired result is reached.

K-means clustering has uses in search engines, market segmentation, statistics and even astronomy. It is used mainly in statistics and can be applied to almost any branch of study. For example, in marketing, it can be used to group different demographics of people into simple groups that make it easier for marketers to target. Astronomers use it to sift through huge amounts of astronomical data; since they cannot analyze each object one by one, they need a way to statistically find points of interest for observation and investigation.

4.8MODULE DESCRIPTION

MODULES

- User interface module
- Login module
- Registration module
- Analysis module
- Report Accident module

User Interface Module

This module provides user an interface to sign up or login into the system. Once the user is logged in he can select Prediction or Analysis or Highway .In each of these

modules the user can select certain attributes and make a prediction or analysis. Once the user has completed the analysis or prediction he can logout.

Login module

When the user of the system encounters the login page, he will be required to enter his username and his password to be able to log into the system. For a login attempt to be successful, the username and password combination input by the user must correspond to values that are available in the database.

Fig.11.Admin Login Page

Registration module

On the Report page, the intended user of the system is required to provide his/her username, password, Full name, phone number and email address. Upon entering these details, the data will be sent into the database. All fields are required for successful registration and if one field is left empty, a prompt will come up.

Analysis module

In this module the user specifies a particular attribute based on which an analysis With respect to its effect on accident can be determined. The analysis will be done using the training data set, Support Vector Machine classifier and the attribute specified by the user. The output will in form of a graph.

Report Accident module

The report page allows the user to interact with the site administrator by reporting the current state of victims and property involved in an accident as he travels along a particular route. This report is therefore sent to the incidence table of the data base, where it is checked and validated by the administrator and then updated into the various fields of the data base for appropriate report generation. The road's users are allowed to also report the accident occurrences on a particular road at any point in time. The road user can also make reports even if he does not have an account. The user clicks on the "Chat with Officer" link on the homepage and starts a chat with an office; he will be required to enter his name, his phone number, e-mail address and the description of the accident. This information is sent to the database where it will be validated by the administrator and finally updated. Below are screenshots of the Road Traffic Accident Monitoring System

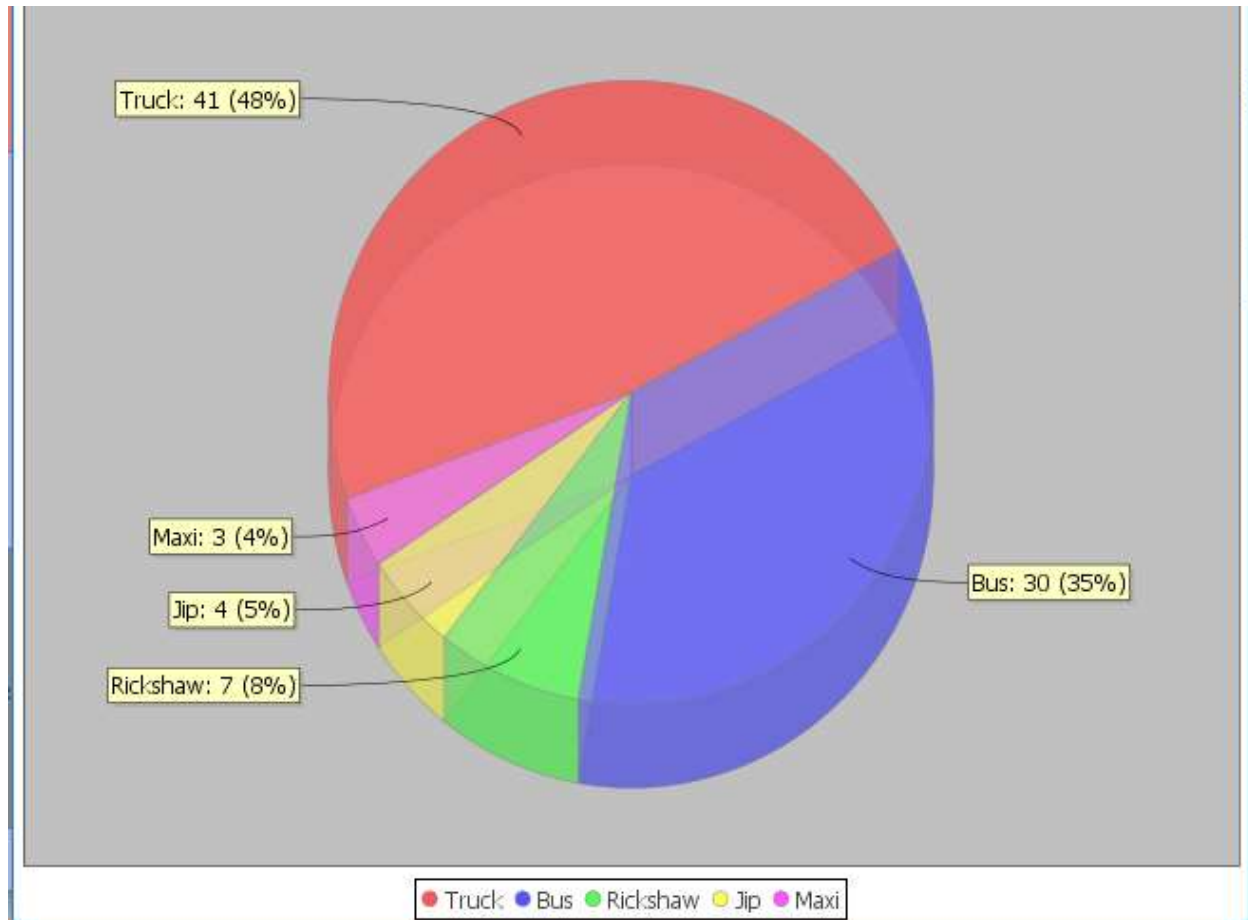


Fig.12. Report Accident module

CHAPTER 5

RESULTS AND DISCUSSION

5.1 PERFORMANCE ANALYSIS

We trained the model for 138 epochs until the training accuracy reaches 94% by specifying the condition to stop training when training accuracy reaches 94%>= using callback function in Tensorflow. The validation steps value is set to 25. The time taken for each epoch is around 32 seconds. The testing accuracy couldn't be displayed along with the training accuracy during training period is because to test for epoch the cost of machine resources increases and due to the dearth of availability of resources (GPU) it evaluated at the end (generally this is the industry

procedure followed for deep learning systems in contrast with the machine learning models).

The training accuracy of the model is 89.281% accuracy which is considered as the final accuracy of the model. The resolution used for the images is 224x224x3 (3 indicates that the image is coloured(RGB)). The loss function used is CategoricalCrossEntropy since this is a multi class classification. RMSprop with the learning rate of 0.0001 (α) is used as an optimizer. ReLU activation function is used for five dense layers other than pretrained model layers and in the last layer softmax activation function is used.

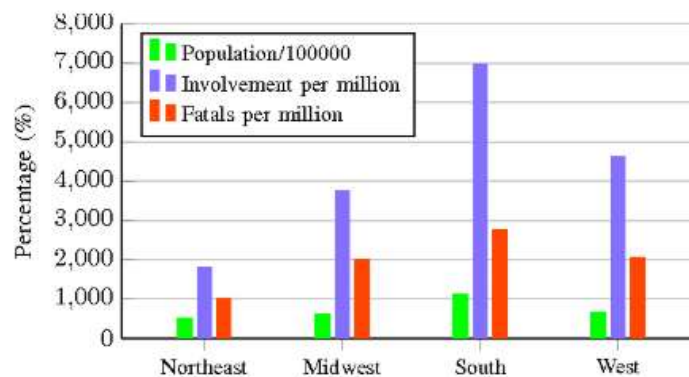


Fig. 18. Gantt Chart

CHAPTER 6

CONCLUSION

1.1 CONCLUSION

As seen in statistics, association rule mining, and the classification, the environmental factors like roadway surface, weather, and light condition do not strongly affect the fatal rate, while the human factors like being drunk or not, and the collision type, have stronger affect on the fatal rate. From the clustering result we could see that some states/regions have higher fatal rate, while some others lower. We may pay more attention when driving within those risky states/regions. Through the task performed, we realized that data seems never to be enough to make a strong decision. If more data,

like non-fatal accident data, weather data, mileage data, and so on, are available, more test could be performed thus more suggestion could be made from the data.

REFERENCES

[1] Divya Bansal and Lekha Bhambhu. Execution of Apriori algorithm of data mining directed towards tumultuous crimes concerning women. International Journal of Advanced Research in Computer Science and Software Engineering, 3(9), September 2013.

Amira A El Tayeb, Vikas Pareek, and Abdelaziz Araar. Applying association rules mining algorithms for traffic accidents in dubai. International Journal of Soft Computing and Engineering, September 2015.

- [3] William M Evanco. The potential impact of rural mayday systems on vehicular crash fatalities. *Accident Analysis & Prevention*, 31(5):455–462, September 1999.
- [4] K Jayasudha and C Chandrasekar. An overview of data mining in road traffic and accident analysis. *Journal of Computer Applications*, 2(4):32–37, 2009.
- [5] S. Krishnaveni and M. Hemalatha. A perspective analysis of traffic accident using data mining techniques. *International Journal of Computer Applications*, 23(7):40–48, June 2011.
- [6] Sachin Kumar and Durga Toshniwal. Analysing road accident data using association rule mining. In *Proceedings of International Conference on Computing, Communication and Security*, pages 1–6, 2015.
- [7] Eric M Ossiander and Peter Cummings. Freeway speed limits and traffic fatalities in washington state. *Accident Analysis & Prevention*, 34(1):13–18, 2002.
- [8] KMA Solaiman, Md Mustafizur Rahman, and Nashid Shahriar. Avra Bangladesh collection, analysis & visualization of road accident data in Bangladesh. In *Proceedings of International Conference on Informatics, Electronics & Vision*, pages 1–6. IEEE, 2013.
- [9] Trac Integrated SCM & Project Management. Fatal Accidents Dataset. <https://wiki.csc.calpoly.edu/datasets/wiki/HighwayAccidents>
- [10] U.S. Census Bureau. Population Estimates. [http://www.census.gov/popest/data/historical/2000s/vintage 2007/](http://www.census.gov/popest/data/historical/2000s/vintage%2007/), 2007.

APPENDICES

A. SOURCE CODE

```
CREATE TABLE IF NOT EXISTS `accident` (  
  `accident_no` int(11) NOT NULL AUTO_INCREMENT,  
  `date` date NOT NULL,  
  `time_id` int(11) NOT NULL,  
  `location_id` int(11) NOT NULL,  
  `type_id` int(11) NOT NULL,  
  `reason_id` int(11) NOT NULL,
```

```

`season_id` int(11) NOT NULL,
`injured` int(11) NOT NULL,
`killed` int(11) NOT NULL,
PRIMARY KEY (`accident_no`),
KEY `time_id` (`time_id`),
KEY `location_id` (`location_id`),
KEY `type_id` (`type_id`),
KEY `reason_id` (`reason_id`),
KEY `season_id` (`season_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=802 ;

```

```
--
```

```
-- Dumping data for table `accident`
```

```
--
```

```

INSERT INTO `accident` (`accident_no`, `date`, `time_id`, `location_id`, `type_id`,
`reason_id`, `season_id`, `injured`, `killed`) VALUES
(1, '2007-05-02', 5, 1, 2, 3, 1, 0, 1),
(2, '2007-05-27', 4, 2, 4, 3, 1, 1, 1),
(3, '2007-06-29', 5, 1, 1, 3, 1, 1, 0),
(4, '2007-07-01', 1, 1, 2, 3, 2, 1, 0),
(5, '2007-08-22', 3, 1, 4, 3, 2, 1, 0),
(6, '2007-09-03', 4, 1, 2, 3, 2, 1, 0),
(7, '2007-09-15', 2, 1, 4, 1, 2, 1, 1),
(8, '2007-10-23', 1, 1, 1, 1, 2, 1, 0),
(9, '2007-11-05', 1, 1, 1, 1, 3, 0, 1),
(10, '2007-11-15', 1, 2, 4, 3, 3, 0, 1),
(11, '2007-12-09', 3, 1, 1, 3, 3, 1, 1),
(12, '2007-12-28', 1, 4, 2, 3, 3, 1, 1),
(13, '2008-03-08', 1, 3, 2, 3, 1, 1, 1),
(14, '2008-04-25', 2, 1, 4, 3, 1, 1, 0),

```

(15, '2008-05-08', 1, 1, 1, 1, 1, 0, 1),
(16, '2008-05-08', 5, 4, 2, 3, 1, 0, 1),
(17, '2008-05-23', 1, 1, 1, 1, 1, 1, 1),
(18, '2008-06-07', 2, 1, 4, 2, 1, 0, 1),
(19, '2008-06-10', 1, 2, 4, 3, 1, 0, 1),
(20, '2008-08-16', 3, 1, 1, 4, 2, 1, 2),
(21, '2008-09-24', 2, 2, 14, 3, 2, 3, 1),
(22, '2008-11-05', 3, 1, 2, 4, 3, 2, 1),
(23, '2009-02-13', 3, 1, 2, 4, 3, 2, 1),
(24, '2009-04-28', 2, 3, 2, 2, 1, 1, 1),
(25, '2009-06-02', 2, 3, 1, 1, 1, 4, 2),
(26, '2009-07-10', 3, 4, 1, 1, 2, 2, 1),
(27, '2009-08-14', 2, 1, 4, 3, 2, 1, 1),
(28, '2009-09-05', 4, 1, 4, 3, 2, 1, 1),
(29, '2009-10-11', 4, 1, 4, 2, 2, 1, 1),
(30, '2009-10-12', 5, 3, 2, 1, 2, 1, 0),
(31, '2009-12-22', 5, 4, 2, 3, 3, 1, 1),
(32, '2010-03-20', 3, 1, 1, 1, 1, 1, 1),
(33, '2010-04-30', 3, 4, 4, 2, 1, 0, 1),
(34, '2010-05-03', 1, 4, 1, 1, 1, 1, 1),
(35, '2010-05-09', 2, 2, 2, 2, 1, 0, 1),
(36, '2010-05-20', 4, 2, 2, 2, 1, 1, 1),
(37, '2010-06-14', 1, 1, 4, 3, 1, 0, 1),
(38, '2010-07-01', 1, 1, 2, 1, 2, 0, 1),
(39, '2010-07-12', 1, 1, 1, 2, 2, 0, 1),
(40, '2010-08-18', 4, 4, 2, 3, 2, 1, 0),
(41, '2010-09-23', 5, 1, 4, 3, 2, 1, 0),
(42, '2011-01-14', 5, 2, 2, 2, 3, 0, 1),
(43, '2011-02-28', 2, 2, 2, 1, 3, 1, 1),
(44, '2011-07-27', 1, 3, 2, 3, 2, 0, 1),
(45, '2011-08-18', 3, 1, 1, 3, 2, 0, 1),

(46, '2011-10-06', 4, 3, 2, 3, 2, 0, 1),
(47, '2011-10-09', 5, 4, 1, 2, 2, 1, 1),
(48, '2011-10-10', 4, 1, 4, 3, 2, 0, 1),
(49, '2011-11-01', 1, 3, 2, 3, 3, 2, 2),
(50, '2011-09-11', 4, 3, 1, 3, 1, 1, 1),
(51, '2011-11-12', 4, 1, 2, 2, 3, 1, 1),
(52, '2011-12-09', 5, 2, 4, 1, 3, 1, 1),
(53, '2011-12-25', 5, 4, 4, 3, 3, 1, 1),
(54, '2012-02-11', 3, 2, 2, 1, 3, 0, 1),
(55, '2012-02-20', 3, 1, 1, 2, 2, 1, 1),
(56, '2012-03-12', 4, 1, 2, 3, 1, 1, 1),
(57, '2012-03-04', 3, 1, 1, 1, 1, 0, 1),
(58, '2012-03-16', 5, 1, 4, 3, 1, 1, 0),
(59, '2012-03-07', 4, 2, 4, 4, 1, 1, 0),
(60, '2012-03-18', 5, 2, 2, 2, 1, 1, 0),
(61, '2012-04-08', 5, 1, 1, 1, 1, 1, 2),
(62, '2012-05-28', 1, 1, 1, 3, 1, 1, 1),
(63, '2012-05-23', 2, 1, 1, 4, 1, 1, 1),
(64, '2012-06-10', 3, 4, 2, 1, 1, 2, 3),
(65, '2012-06-18', 3, 1, 4, 4, 1, 1, 0),
(66, '2012-07-11', 4, 1, 2, 1, 2, 0, 1),
(67, '2012-08-02', 3, 2, 2, 2, 2, 0, 1),
(68, '2012-08-21', 1, 1, 4, 3, 2, 1, 1),
(69, '2012-08-22', 2, 2, 1, 2, 2, 1, 0),
(70, '2012-08-29', 4, 1, 2, 3, 2, 1, 0),
(71, '2012-09-04', 1, 2, 1, 2, 2, 1, 0),
(72, '2012-09-22', 3, 3, 2, 2, 2, 1, 0),
(73, '2012-10-12', 3, 1, 4, 3, 2, 0, 2),
(74, '2012-10-31', 3, 2, 1, 3, 2, 1, 0),
(75, '2012-11-05', 5, 1, 2, 3, 3, 1, 0),
(76, '2012-11-19', 4, 1, 2, 1, 3, 1, 1),

(77, '2012-12-05', 5, 1, 1, 1, 3, 1, 1),
(78, '2012-12-10', 5, 2, 1, 1, 3, 1, 1),
(79, '2012-12-26', 5, 1, 1, 1, 2, 3, 2),
(80, '2013-02-01', 3, 3, 2, 4, 3, 1, 1),
(81, '2013-02-14', 3, 2, 8, 3, 3, 1, 1),
(82, '2013-02-20', 2, 1, 2, 4, 3, 1, 0),
(83, '2012-02-14', 5, 2, 1, 4, 3, 1, 0),
(84, '2013-03-21', 3, 1, 4, 2, 1, 2, 2),
(85, '2013-03-30', 4, 4, 5, 4, 1, 0, 1),
(86, '2013-04-13', 4, 3, 2, 1, 1, 1, 1),
(87, '2013-04-24', 5, 3, 1, 2, 1, 0, 1),
(88, '2013-05-11', 1, 1, 2, 2, 1, 2, 0),
(89, '2013-05-30', 3, 3, 5, 4, 3, 2, 0),
(90, '2013-06-15', 3, 2, 2, 4, 1, 0, 2),
(91, '2013-06-13', 4, 4, 8, 3, 1, 3, 2),
(92, '2013-06-30', 5, 4, 1, 1, 1, 1, 1),
(93, '2013-07-11', 3, 1, 1, 1, 2, 1, 1),
(94, '2013-07-26', 1, 4, 9, 4, 3, 1, 1),
(95, '2013-08-02', 1, 2, 9, 3, 2, 2, 5),
(96, '2013-07-19', 5, 1, 3, 1, 2, 1, 1),
(97, '2013-08-09', 5, 4, 3, 2, 2, 2, 1),
(98, '2013-08-30', 5, 3, 8, 1, 2, 2, 2),
(99, '2013-09-07', 1, 2, 1, 1, 2, 0, 1),
(100, '2013-10-11', 3, 1, 11, 4, 2, 2, 0),
(101, '2013-11-02', 4, 3, 1, 3, 2, 1, 1),
(102, '2013-11-30', 3, 2, 1, 2, 3, 1, 0),
(103, '2013-11-30', 2, 3, 8, 4, 3, 1, 1),
(104, '2013-12-11', 4, 3, 2, 2, 3, 0, 1),
(105, '2013-12-27', 5, 2, 1, 1, 2, 1, 1),
(106, '2014-01-01', 5, 5, 4, 3, 3, 3, 0),
(107, '2014-01-22', 5, 3, 7, 4, 3, 0, 2),

(108, '2014-01-31', 1, 3, 1, 1, 3, 1, 1),
(109, '2014-02-02', 4, 3, 1, 1, 3, 1, 1),
(110, '2014-02-28', 1, 2, 6, 2, 3, 0, 1),
(111, '2014-03-11', 1, 1, 2, 1, 3, 2, 2),
(112, '2014-03-01', 2, 4, 5, 4, 3, 3, 0),
(113, '2014-03-05', 2, 4, 1, 2, 1, 1, 0),
(114, '2014-03-13', 4, 3, 5, 4, 1, 3, 0),
(115, '2014-03-24', 4, 2, 5, 3, 1, 1, 0),
(116, '2014-05-14', 5, 4, 3, 1, 1, 1, 1),
(117, '2014-06-03', 2, 1, 1, 1, 1, 1, 1),
(118, '2014-08-04', 1, 4, 4, 2, 2, 1, 1),
(119, '2014-09-23', 2, 2, 2, 2, 2, 1, 1),
(120, '2014-09-25', 1, 1, 1, 1, 2, 4, 0),
(121, '2014-11-10', 3, 1, 1, 4, 3, 0, 1),
(122, '2014-11-19', 2, 1, 1, 3, 3, 3, 0),
(123, '2014-12-06', 3, 1, 5, 3, 3, 3, 0),
(124, '2014-12-22', 2, 2, 3, 3, 3, 3, 0),
(125, '2014-12-30', 3, 2, 3, 3, 3, 3, 0),
(126, '2014-12-31', 1, 1, 1, 1, 3, 2, 5),
(127, '2015-01-04', 3, 3, 4, 1, 3, 3, 1),
(128, '2015-01-05', 4, 2, 1, 4, 3, 1, 0),
(129, '2015-01-13', 5, 1, 1, 1, 3, 2, 2),
(130, '2015-02-14', 1, 1, 3, 3, 3, 0, 1),
(131, '2015-02-27', 5, 3, 4, 4, 3, 2, 1),
(132, '2015-03-12', 4, 1, 1, 1, 1, 1, 0),
(133, '2015-03-18', 4, 4, 3, 3, 1, 3, 1),
(134, '2015-03-20', 3, 3, 2, 2, 1, 1, 1),
(135, '2015-03-30', 3, 2, 2, 3, 1, 1, 1),
(136, '2015-04-05', 5, 1, 5, 4, 1, 1, 0),
(137, '2015-04-16', 1, 3, 6, 3, 1, 1, 0),
(138, '2015-04-22', 4, 4, 4, 2, 1, 2, 1),

(139, '2015-04-24', 1, 1, 1, 1, 1, 1, 1),
(140, '2015-04-01', 1, 1, 2, 2, 1, 3, 1),
(141, '2015-04-30', 3, 4, 3, 3, 1, 1, 1),
(142, '2015-05-21', 1, 2, 2, 3, 1, 3, 1),
(144, '2015-05-25', 4, 4, 1, 1, 1, 1, 1),
(145, '2015-05-28', 4, 2, 2, 2, 1, 2, 2),
(146, '2015-05-30', 3, 3, 2, 2, 1, 1, 1),
(147, '2015-06-04', 3, 4, 4, 3, 1, 1, 1),
(148, '2015-06-09', 2, 4, 4, 4, 1, 3, 1),
(149, '2015-06-11', 1, 1, 1, 1, 1, 1, 1),
(150, '2015-06-22', 1, 1, 2, 4, 1, 1, 0),
(151, '2015-06-13', 3, 3, 3, 3, 1, 1, 1),
(152, '2015-06-30', 3, 1, 1, 4, 1, 1, 1),
(153, '2015-08-03', 5, 2, 1, 3, 2, 2, 2),
(154, '2015-08-16', 5, 4, 3, 4, 2, 4, 2),
(155, '2015-08-30', 4, 2, 5, 3, 1, 1, 1),
(156, '2015-09-12', 4, 3, 3, 2, 2, 1, 1),
(157, '2015-09-22', 2, 2, 2, 4, 2, 1, 1),
(158, '2015-09-30', 4, 3, 2, 2, 2, 1, 0),
(159, '2015-10-03', 4, 3, 6, 4, 1, 1, 1),
(160, '2015-10-12', 4, 3, 4, 4, 2, 1, 1),
(161, '2015-10-15', 5, 4, 4, 3, 2, 1, 1),
(162, '2015-10-17', 1, 1, 1, 1, 2, 2, 1),
(163, '2015-10-30', 4, 1, 1, 1, 2, 3, 0),
(164, '2015-11-16', 1, 3, 2, 2, 2, 1, 1),
(165, '2015-11-17', 4, 2, 2, 1, 3, 2, 2),
(166, '2016-04-26', 3, 1, 2, 1, 1, 1, 1),
(167, '2016-05-05', 2, 4, 4, 3, 1, 1, 0),
(168, '2016-06-16', 4, 4, 1, 3, 1, 1, 0),
(169, '2022-05-16', 3, 4, 2, 3, 1, 1, 1),
(170, '2016-06-18', 1, 2, 2, 2, 1, 1, 0),

(171, '2016-06-30', 2, 2, 2, 1, 1, 3, 1),
(172, '2016-07-13', 3, 2, 2, 1, 2, 1, 1),
(176, '2016-07-16', 1, 4, 2, 3, 2, 1, 1),
(177, '2016-08-22', 3, 1, 2, 2, 2, 1, 0),
(178, '2016-09-06', 2, 2, 2, 2, 2, 1, 0),
(179, '2016-10-04', 1, 2, 4, 2, 2, 1, 0),
(180, '2007-01-12', 3, 4, 2, 3, 3, 1, 0),
(181, '2007-01-15', 3, 6, 4, 3, 3, 1, 0),
(182, '2007-02-14', 3, 7, 1, 2, 3, 2, 1),
(183, '2007-02-17', 3, 7, 1, 2, 3, 5, 0),
(184, '2007-03-01', 4, 5, 4, 2, 1, 2, 1),
(185, '2007-05-17', 5, 6, 4, 4, 1, 1, 0),
(186, '2007-06-01', 4, 6, 1, 1, 1, 0, 1),
(187, '2007-09-13', 5, 7, 3, 3, 2, 1, 0),
(188, '2007-11-16', 3, 5, 1, 3, 3, 2, 1),
(189, '2008-01-01', 4, 6, 2, 3, 3, 1, 0),
(190, '2008-02-16', 4, 5, 2, 3, 3, 1, 0),
(191, '2008-04-03', 4, 5, 2, 3, 1, 2, 1),
(192, '2008-05-13', 4, 5, 7, 2, 1, 1, 0),
(193, '2008-05-30', 5, 7, 3, 3, 1, 1, 0),
(194, '2008-07-12', 3, 7, 2, 3, 2, 2, 0),
(195, '2008-08-01', 1, 6, 7, 2, 2, 1, 1),
(196, '2008-09-12', 3, 7, 4, 1, 2, 2, 1),
(197, '2008-10-14', 2, 5, 7, 3, 2, 1, 1),
(198, '2008-11-21', 5, 7, 2, 1, 3, 1, 1),
(199, '2008-12-01', 3, 6, 1, 2, 3, 0, 1),
(200, '2008-12-30', 4, 6, 9, 4, 2, 1, 1),
(201, '2009-01-14', 4, 7, 1, 1, 3, 10, 0),
(202, '2009-07-03', 4, 6, 7, 1, 2, 2, 2),
(203, '2009-09-12', 3, 5, 1, 1, 2, 10, 2),
(204, '2009-09-26', 5, 7, 8, 4, 2, 1, 0),

(205, '2009-01-14', 5, 5, 7, 4, 3, 1, 0),
(206, '2010-05-03', 5, 5, 1, 3, 1, 1, 0),
(207, '2010-07-18', 4, 6, 9, 4, 2, 1, 0),
(208, '2010-08-18', 1, 2, 2, 4, 2, 1, 0),
(209, '2011-01-14', 3, 6, 11, 2, 3, 3, 0),
(210, '2011-03-14', 4, 5, 12, 1, 1, 2, 1),
(211, '2011-04-01', 5, 7, 1, 3, 1, 1, 1),
(212, '2011-08-11', 4, 6, 4, 3, 2, 1, 1),
(213, '2011-11-15', 5, 5, 11, 1, 2, 3, 3),
(214, '2012-03-16', 2, 7, 8, 2, 1, 8, 1),
(215, '2012-03-19', 3, 5, 3, 4, 3, 2, 1),
(216, '2012-04-18', 2, 7, 1, 1, 1, 1, 1),
(217, '2012-05-19', 1, 5, 2, 2, 1, 3, 1),
(218, '2012-07-11', 4, 7, 1, 1, 2, 5, 1),
(219, '2012-08-21', 5, 6, 10, 1, 2, 1, 1),
(220, '2012-09-22', 4, 5, 4, 3, 2, 1, 2),
(221, '2012-10-13', 5, 6, 8, 2, 2, 3, 2),
(222, '2012-11-01', 1, 7, 1, 3, 3, 0, 1),
(223, '2012-11-30', 3, 5, 7, 4, 3, 1, 1),
(224, '2012-12-17', 3, 7, 2, 3, 3, 1, 1),
(225, '2013-01-14', 5, 5, 2, 1, 3, 1, 0),
(226, '2013-03-14', 1, 7, 14, 1, 1, 2, 1),
(227, '2013-04-18', 5, 7, 6, 3, 1, 1, 1),
(228, '2013-07-09', 5, 6, 7, 1, 2, 3, 0),
(229, '2013-11-10', 4, 7, 8, 2, 3, 2, 0),
(230, '2013-12-23', 1, 5, 9, 1, 3, 2, 0),
(231, '2014-04-14', 5, 6, 1, 1, 1, 2, 0),
(232, '2014-04-30', 2, 7, 2, 2, 1, 1, 1),
(233, '2014-05-21', 1, 5, 9, 2, 1, 1, 1),
(234, '2014-06-01', 4, 6, 2, 1, 1, 3, 2),
(235, '2014-07-11', 1, 7, 6, 2, 2, 4, 1),

(236, '2014-08-11', 1, 7, 1, 2, 3, 1, 2),
(237, '2014-11-01', 1, 7, 1, 1, 3, 1, 1),
(238, '2014-12-03', 3, 6, 2, 2, 3, 2, 2),
(239, '2015-01-30', 1, 5, 8, 3, 3, 1, 0),
(240, '2015-02-01', 1, 5, 2, 2, 3, 2, 1),
(241, '2015-02-27', 1, 7, 1, 3, 3, 1, 1),
(242, '2015-03-04', 4, 6, 7, 1, 1, 1, 1),
(243, '2015-04-11', 2, 5, 10, 1, 1, 1, 1),
(244, '2015-05-11', 4, 6, 9, 1, 1, 0, 1),
(245, '2015-07-11', 1, 7, 1, 2, 2, 2, 2),
(246, '2015-08-02', 3, 7, 2, 1, 2, 1, 1),
(247, '2015-10-19', 1, 6, 1, 2, 2, 2, 2),
(248, '2015-11-09', 1, 5, 11, 2, 3, 1, 3),
(249, '2015-12-31', 5, 5, 14, 1, 3, 1, 2),
(250, '2016-01-20', 5, 5, 2, 3, 3, 3, 1),
(251, '2016-02-23', 3, 6, 8, 1, 3, 1, 1),
(252, '2016-03-15', 4, 5, 9, 4, 1, 1, 0),
(253, '2016-05-20', 1, 6, 1, 2, 1, 1, 1),
(254, '2016-07-01', 2, 7, 10, 1, 2, 1, 0),
(255, '2016-08-18', 5, 7, 6, 4, 2, 1, 3),
(256, '2016-09-30', 2, 6, 4, 1, 2, 1, 1),
(257, '2016-10-30', 5, 5, 5, 1, 2, 1, 1),
(258, '2016-11-21', 4, 5, 11, 1, 3, 2, 1),
(259, '2016-12-31', 5, 7, 2, 3, 3, 1, 1),
(260, '2007-01-23', 5, 9, 2, 3, 3, 0, 1),
(261, '2007-01-25', 5, 9, 2, 3, 3, 1, 1),
(262, '2007-02-28', 1, 8, 5, 3, 2, 1, 1),
(263, '2007-03-12', 3, 8, 9, 1, 1, 1, 2),
(264, '2007-04-11', 1, 12, 6, 4, 1, 1, 2),
(265, '2007-04-13', 3, 8, 2, 1, 1, 2, 1),
(266, '2007-05-04', 1, 8, 6, 1, 1, 2, 1),

(267, '2007-05-05', 5, 11, 4, 4, 1, 2, 0),
(268, '2007-05-15', 3, 10, 2, 3, 1, 1, 1),
(269, '2007-05-21', 5, 12, 10, 2, 1, 1, 3),
(270, '2007-05-30', 2, 9, 7, 4, 1, 2, 1),
(271, '2007-06-01', 1, 8, 1, 2, 1, 1, 1),
(272, '2007-07-10', 4, 9, 5, 2, 2, 1, 1),
(273, '2007-07-10', 5, 11, 11, 4, 2, 2, 1),
(274, '2007-08-01', 2, 11, 9, 1, 2, 1, 2),
(275, '2007-08-10', 2, 8, 7, 2, 2, 1, 1),
(276, '2007-08-11', 2, 8, 5, 3, 1, 1, 2),
(277, '2007-08-21', 3, 8, 2, 1, 2, 1, 1),
(278, '2007-08-27', 5, 10, 10, 3, 2, 1, 1),
(279, '2007-08-30', 1, 7, 5, 1, 2, 1, 3),
(280, '2007-09-12', 2, 8, 2, 1, 2, 1, 1),
(281, '2007-09-11', 4, 8, 5, 1, 2, 1, 1),
(282, '2007-09-20', 5, 12, 6, 3, 1, 1, 1),
(283, '2007-10-19', 2, 11, 1, 3, 2, 2, 1),
(284, '2007-10-29', 1, 13, 13, 1, 2, 3, 0),
(285, '2007-10-30', 5, 8, 5, 2, 2, 2, 2),
(287, '2007-10-31', 5, 9, 6, 2, 2, 1, 2),
(288, '2007-11-01', 3, 10, 10, 1, 1, 1, 1),
(289, '2007-11-03', 3, 8, 11, 3, 3, 3, 0),
(290, '2007-11-17', 2, 9, 6, 3, 3, 1, 1),
(291, '2007-11-23', 4, 9, 1, 1, 1, 1, 1),
(292, '2007-11-30', 4, 8, 11, 1, 3, 1, 1),
(293, '2007-12-05', 5, 9, 7, 1, 3, 1, 0),
(294, '2007-12-14', 5, 8, 6, 2, 3, 1, 1),
(295, '2007-12-25', 5, 11, 2, 3, 3, 1, 1),
(296, '2008-01-02', 1, 10, 2, 2, 3, 1, 1),
(297, '2008-01-10', 3, 11, 3, 3, 3, 2, 1),
(298, '2008-01-14', 5, 8, 6, 4, 3, 4, 1),

(299, '2008-01-19', 4, 9, 13, 1, 3, 1, 0),
(300, '2008-01-28', 3, 8, 1, 1, 3, 3, 1),
(301, '2008-01-31', 5, 9, 12, 2, 3, 0, 2),
(302, '2008-02-02', 3, 10, 4, 3, 3, 2, 1),
(303, '2008-02-05', 5, 10, 2, 2, 3, 1, 1),
(304, '2008-02-12', 1, 3, 5, 1, 3, 2, 1),
(305, '2008-02-14', 4, 9, 6, 1, 3, 4, 0),
(306, '2008-02-20', 2, 8, 7, 4, 3, 1, 1),
(307, '2008-02-28', 3, 11, 12, 3, 3, 1, 1),
(308, '2008-02-28', 1, 12, 8, 1, 3, 3, 1),
(309, '2008-03-04', 5, 13, 14, 4, 3, 1, 1),
(310, '2008-03-08', 2, 9, 1, 2, 1, 1, 1),
(311, '2008-03-17', 5, 8, 5, 2, 1, 2, 1),
(312, '2008-03-16', 1, 13, 13, 3, 1, 2, 1),
(313, '2008-03-25', 4, 11, 2, 1, 1, 1, 1),
(314, '2008-03-21', 5, 8, 4, 3, 1, 1, 1),
(315, '2008-03-28', 1, 9, 9, 2, 1, 1, 1),
(316, '2008-03-30', 5, 11, 12, 1, 1, 3, 0),
(317, '2008-04-02', 1, 13, 3, 2, 1, 1, 0),
(318, '2008-04-06', 1, 10, 3, 1, 1, 1, 0),
(319, '2008-04-08', 5, 12, 12, 3, 1, 1, 1),
(320, '2008-04-09', 4, 9, 1, 2, 1, 1, 3),
(321, '2008-04-20', 5, 9, 2, 2, 1, 2, 0),
(322, '2008-04-27', 2, 12, 12, 4, 1, 1, 0),
(323, '2008-04-30', 4, 8, 6, 4, 1, 3, 0),
(324, '2008-05-04', 5, 8, 5, 1, 1, 2, 1),
(325, '2008-05-12', 5, 9, 4, 3, 1, 3, 1),
(326, '2008-05-18', 5, 9, 1, 1, 1, 3, 1),
(327, '2008-05-17', 5, 9, 13, 2, 1, 1, 1),
(328, '2008-05-18', 4, 10, 4, 1, 1, 1, 1),
(329, '2008-05-20', 5, 9, 10, 1, 1, 1, 1),

(330, '2008-06-01', 2, 11, 2, 2, 1, 1, 0),
(331, '2008-06-13', 4, 9, 9, 1, 1, 4, 0),
(332, '2008-06-29', 2, 8, 8, 4, 1, 2, 1),
(333, '2008-07-12', 2, 13, 13, 2, 2, 1, 1),
(334, '2008-07-20', 3, 10, 10, 4, 3, 1, 1),
(335, '2008-07-30', 1, 12, 2, 1, 2, 1, 1),
(336, '2008-07-30', 2, 11, 5, 3, 1, 1, 0),
(337, '2008-08-06', 2, 13, 14, 1, 2, 3, 1),
(338, '2008-08-15', 3, 13, 14, 1, 1, 1, 1),
(339, '2008-09-15', 1, 13, 12, 2, 2, 1, 1),
(340, '2008-09-17', 4, 13, 14, 2, 2, 3, 0),
(341, '2008-09-29', 3, 11, 2, 1, 2, 1, 2),
(342, '2008-09-30', 4, 8, 10, 2, 1, 2, 0),
(343, '2008-09-30', 5, 10, 10, 1, 2, 1, 1),
(344, '2008-10-11', 4, 11, 12, 1, 1, 1, 0),
(345, '2008-10-21', 2, 8, 11, 2, 1, 1, 1),
(346, '2008-10-27', 1, 11, 12, 3, 2, 1, 1),
(347, '2008-10-16', 1, 8, 1, 2, 1, 2, 1),
(348, '2008-10-30', 2, 5, 5, 2, 2, 1, 0),
(349, '2008-10-30', 2, 9, 2, 1, 2, 1, 1),
(350, '2008-11-03', 2, 9, 9, 1, 2, 1, 1),
(351, '2008-11-17', 2, 9, 8, 1, 2, 1, 1),
(352, '2008-11-12', 4, 10, 10, 3, 2, 1, 1),
(353, '2008-11-21', 3, 10, 10, 2, 1, 2, 1),
(354, '2008-11-24', 1, 9, 1, 3, 2, 1, 1),
(355, '2008-11-30', 1, 11, 11, 2, 1, 1, 1),
(356, '2008-11-30', 4, 11, 13, 1, 2, 1, 1),
(357, '2008-12-14', 2, 10, 11, 2, 2, 0, 1),
(358, '2008-12-15', 3, 10, 10, 1, 2, 1, 1),
(359, '2008-12-20', 1, 13, 14, 2, 1, 2, 1),
(360, '2008-12-31', 3, 2, 2, 1, 3, 1, 2),

(361, '2009-06-03', 2, 11, 2, 1, 3, 1, 2),
(362, '2009-03-25', 3, 11, 1, 2, 1, 1, 0),
(363, '2009-01-02', 2, 12, 10, 2, 1, 2, 1),
(364, '2009-01-10', 1, 13, 2, 1, 2, 1, 1),
(365, '2009-01-17', 2, 13, 10, 2, 1, 2, 1),
(366, '2009-01-25', 4, 11, 10, 2, 1, 2, 1),
(367, '2009-01-31', 4, 9, 1, 2, 1, 2, 1),
(368, '2009-02-14', 3, 8, 1, 2, 3, 2, 1),
(369, '2009-02-11', 3, 11, 3, 1, 3, 1, 1),
(370, '2009-02-22', 3, 11, 12, 2, 3, 1, 1),
(371, '2009-02-22', 4, 9, 11, 2, 3, 3, 1),
(372, '2009-02-10', 3, 9, 5, 1, 3, 1, 1),
(373, '2009-02-28', 3, 10, 1, 3, 3, 1, 1),
(374, '2009-03-12', 4, 11, 11, 1, 1, 1, 1),
(375, '2009-03-14', 2, 9, 9, 1, 1, 1, 0),
(376, '2009-03-16', 3, 10, 10, 1, 3, 2, 1),
(377, '2009-03-22', 4, 11, 11, 1, 1, 1, 0),
(378, '2009-03-16', 5, 11, 5, 1, 1, 1, 1),
(379, '2009-03-21', 5, 8, 5, 4, 1, 1, 1),
(380, '2009-03-23', 5, 10, 2, 1, 1, 1, 1),
(381, '2009-03-25', 1, 11, 11, 1, 1, 1, 1),
(382, '2009-03-27', 2, 9, 9, 1, 1, 3, 1),
(383, '2009-03-29', 1, 9, 7, 2, 1, 2, 1),
(384, '2009-03-30', 2, 9, 6, 1, 1, 2, 0),
(385, '2009-03-31', 4, 8, 7, 1, 1, 1, 3),
(386, '2009-04-01', 5, 12, 12, 4, 1, 4, 0),
(387, '2009-04-02', 5, 11, 12, 1, 1, 0, 1),
(388, '2009-04-12', 1, 11, 10, 1, 1, 1, 0),
(389, '2009-04-19', 5, 9, 2, 2, 1, 2, 1),
(390, '2009-04-19', 2, 12, 12, 1, 1, 1, 1),
(391, '2009-04-20', 2, 12, 12, 1, 2, 1, 1),

(392, '2009-04-22', 3, 12, 11, 1, 1, 1, 1),
(393, '2009-04-30', 3, 10, 10, 1, 1, 2, 1),
(394, '2009-05-02', 1, 12, 12, 1, 1, 1, 1),
(395, '2009-05-12', 4, 12, 1, 2, 1, 1, 0),
(396, '2009-05-13', 5, 11, 11, 3, 1, 2, 1),
(397, '2009-05-15', 5, 8, 5, 4, 1, 1, 1),
(398, '2009-05-22', 4, 13, 13, 2, 1, 2, 0),
(399, '2009-05-20', 4, 11, 2, 1, 1, 2, 1),
(400, '2009-05-21', 2, 11, 12, 2, 1, 2, 1),
(401, '2009-05-22', 4, 11, 6, 1, 1, 1, 1),
(402, '2009-05-24', 5, 8, 2, 3, 1, 1, 1),
(403, '2009-05-27', 4, 11, 11, 1, 1, 1, 1),
(404, '2009-05-29', 2, 12, 12, 1, 1, 1, 1),
(405, '2009-05-29', 3, 12, 12, 1, 3, 2, 1),
(406, '2009-05-30', 5, 11, 11, 2, 1, 2, 1),
(407, '2009-05-31', 5, 11, 9, 1, 1, 1, 1),
(408, '2009-06-11', 2, 8, 9, 2, 1, 3, 1),
(409, '2009-06-12', 5, 9, 9, 4, 2, 1, 2),
(410, '2009-06-18', 5, 13, 14, 2, 1, 2, 1),
(411, '2009-06-21', 3, 14, 14, 1, 1, 1, 1),
(412, '2009-06-25', 5, 9, 8, 2, 1, 2, 1),
(413, '2009-06-29', 5, 13, 12, 1, 1, 2, 1),
(414, '2009-06-30', 5, 11, 11, 2, 1, 2, 1),
(415, '2009-07-03', 5, 9, 2, 3, 1, 3, 1),
(416, '2009-07-06', 5, 13, 2, 2, 2, 2, 1),
(417, '2009-07-12', 1, 13, 11, 1, 2, 2, 1),
(418, '2009-07-14', 5, 9, 11, 2, 1, 2, 1),
(419, '2009-07-17', 4, 12, 11, 2, 1, 2, 1),
(420, '2009-07-11', 2, 13, 11, 2, 2, 1, 1),
(421, '2009-07-22', 1, 9, 1, 2, 1, 2, 1),
(422, '2009-07-27', 1, 14, 11, 2, 2, 1, 2),

(423, '2009-09-12', 5, 11, 12, 2, 1, 2, 1),
(424, '2009-09-22', 4, 11, 12, 1, 2, 1, 2),
(425, '2009-09-27', 5, 11, 12, 3, 2, 2, 1),
(426, '2009-09-30', 5, 9, 14, 1, 2, 1, 1),
(427, '2009-10-05', 4, 11, 12, 2, 2, 2, 1),
(428, '2009-10-11', 3, 1, 2, 4, 2, 1, 1),
(429, '2009-10-15', 3, 8, 2, 1, 2, 1, 1),
(430, '2009-10-15', 5, 11, 12, 1, 2, 1, 0),
(431, '2009-10-17', 5, 11, 12, 1, 2, 1, 1),
(432, '2009-10-28', 1, 11, 11, 2, 2, 2, 1),
(433, '2009-10-31', 2, 11, 12, 2, 2, 1, 2),
(434, '2009-10-31', 3, 13, 14, 1, 2, 1, 2),
(435, '2009-11-10', 5, 11, 12, 2, 2, 1, 1),
(436, '2009-11-04', 4, 11, 14, 2, 2, 2, 1),
(437, '2009-11-10', 4, 11, 12, 1, 2, 1, 2),
(438, '2010-01-01', 2, 13, 3, 2, 3, 1, 3),
(439, '2010-01-04', 4, 8, 12, 2, 1, 2, 1),
(440, '2009-01-12', 4, 11, 12, 1, 2, 1, 2),
(441, '2010-01-12', 4, 11, 12, 2, 1, 2, 1),
(442, '2010-01-10', 3, 11, 12, 1, 2, 1, 2),
(443, '2010-01-18', 5, 11, 11, 4, 2, 3, 1),
(444, '2010-01-11', 5, 11, 10, 2, 3, 3, 1),
(445, '2010-01-12', 2, 12, 11, 1, 2, 2, 1),
(446, '2010-01-13', 5, 11, 12, 2, 3, 2, 1),
(447, '2010-02-02', 5, 11, 12, 1, 3, 2, 1),
(448, '2010-02-14', 5, 11, 12, 3, 3, 2, 1),
(449, '2010-02-15', 5, 13, 11, 3, 2, 1, 1),
(450, '2010-02-16', 5, 11, 3, 1, 3, 1, 2),
(451, '2010-02-28', 3, 11, 12, 1, 3, 3, 1),
(452, '2010-03-04', 3, 12, 5, 1, 1, 1, 0),
(453, '2010-03-12', 5, 8, 8, 2, 1, 2, 1),

(454, '2010-03-19', 4, 11, 7, 2, 1, 1, 2),
(455, '2010-03-23', 5, 9, 7, 2, 1, 2, 1),
(456, '2010-03-30', 5, 10, 11, 3, 1, 2, 1),
(457, '2010-04-05', 1, 9, 10, 2, 1, 2, 1),
(458, '2010-04-10', 4, 9, 6, 2, 1, 3, 1),
(459, '2010-04-19', 5, 11, 10, 2, 1, 2, 1),
(460, '2010-04-26', 2, 9, 9, 2, 1, 1, 0),
(461, '2010-04-30', 1, 11, 1, 2, 1, 1, 1),
(462, '2010-05-04', 4, 13, 2, 3, 1, 2, 1),
(463, '2010-05-11', 2, 10, 11, 3, 1, 2, 0),
(464, '2010-05-19', 1, 12, 11, 2, 1, 2, 1),
(465, '2010-05-20', 3, 12, 4, 2, 1, 2, 0),
(466, '2010-05-29', 5, 9, 2, 3, 2, 1, 1),
(467, '2010-06-04', 3, 10, 11, 2, 1, 2, 0),
(468, '2010-06-14', 1, 9, 9, 2, 1, 2, 0),
(469, '2010-06-20', 4, 8, 1, 2, 1, 2, 1),
(470, '2010-07-12', 5, 9, 2, 3, 2, 1, 1),
(471, '2010-07-18', 2, 10, 11, 1, 2, 1, 1),
(472, '2010-07-28', 5, 11, 11, 2, 2, 3, 0),
(473, '2010-07-30', 5, 11, 12, 1, 2, 1, 1),
(474, '2010-08-13', 1, 11, 12, 1, 2, 1, 1),
(475, '2010-09-11', 2, 11, 8, 1, 2, 2, 1),
(476, '2010-11-10', 3, 9, 12, 1, 3, 1, 2),
(477, '2010-12-11', 2, 7, 11, 1, 2, 1, 1),
(478, '2010-08-11', 5, 12, 11, 1, 2, 1, 2),
(479, '2010-12-11', 5, 13, 11, 1, 2, 1, 1),
(480, '2010-12-21', 5, 8, 10, 1, 3, 1, 1),
(481, '2011-01-02', 4, 14, 7, 3, 3, 3, 0),
(482, '2011-01-13', 1, 17, 8, 1, 3, 4, 1),
(483, '2011-01-19', 4, 15, 1, 1, 3, 8, 2),
(484, '2011-02-03', 4, 17, 3, 1, 3, 5, 1),

```
(485, '2011-02-13', 2, 14, 6, 2, 3, 3, 0),
(486, '2011-02-21', 3, 16, 7, 3, 3, 2, 0),
(487, '2011-03-04', 5, 14, 2, 1, 1, 3, 1),
(488, '2011-03-10', 4, 18, 2, 3, 1, 2, 0),
(489, '2011-03-15', 1, 14, 1, 2, 1, 7, 2),
(490, '2011-03-31', 3, 15, 9, 2, 1, 3, 1),
(491, '2011-04-06', 5, 15, 2, 4, 1, 3, 0),
(492, '2011-04-17', 3, 18, 6, 3, 1, 4, 1),
(493, '2011-04-22', 2, 17, 2, 1, 1, 3, 0),
(494, '2011-05-11', 4, 14, 10, 1, 1, 2, 0),
(495, '2011-05-13', 1, 17, 9, 2, 1, 3, 1),
(496, '2011-05-26', 5, 14, 2, 2, 1, 2, 1),
(497, '2011-06-01', 3, 15, 5, 1, 1, 7, 1),
(498, '2011-06-14', 3, 16, 12, 3, 1, 2, 1),
(499, '2011-06-30', 4, 18, 2, 2, 1, 3, 0),
(500, '2011-07-07', 2, 14, 1, 3, 2, 7, 1),
```

```
-- -----
```

```
--
```

```
-- Table structure for table `day_time`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `day_time` (
  `time_id` int(11) NOT NULL AUTO_INCREMENT,
  `time` varchar(30) NOT NULL,
  PRIMARY KEY (`time_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```

```
--
```

```
-- Dumping data for table `day_time`
```

```
--
```

```
INSERT INTO `day_time` (`time_id`, `time`) VALUES
```

```
(1, 'Morning'),
```

```
(2, 'Noon'),
```

```
(3, 'Afternoon'),
```

```
(4, 'Evening'),
```

```
(5, 'Night');
```

```
-- -----
```

```
--
```

```
-- Table structure for table `location`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `location` (
```

```
  `location_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `location_name` varchar(30) NOT NULL,
```

```
  `lat` double NOT NULL,
```

```
  `lon` double NOT NULL,
```

```
  PRIMARY KEY (`location_id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;
```

```
--
```

```
-- Dumping data for table `location`
```

```
--
```

```
INSERT INTO `location` (`location_id`, `location_name`, `lat`, `lon`) VALUES
```

```
(1, 'Mohipal', 23.009, 91.382),
```

```
(2, 'Lalpol', 22.983, 91.412),
```

(3, 'East Chilonia', 22.9668, 91.44577),
 (4, 'Lemua', 22.95954, 91.43957),
 (5, 'Koska', 22.9492, 91.46403),
 (6, 'Fazilpur', 22.9548, 91.48661),
 (7, 'Muhurigonj', 22.94498, 91.5102),
 (8, 'Braraiyarhat', 22.8955, 91.5325),
 (9, 'Mithachara', 22.8048, 91.557),
 (10, 'Mirsharai', 22.77396, 91.57297),
 (11, 'Noyduaria', 22.7374, 91.5971),
 (12, 'Nizampur', 22.718, 91.6056),
 (13, 'Borodarogarhat', 22.6778, 91.62616),
 (14, 'Panthasala', 22.6372, 91.6476),
 (15, 'Barabkunda', 22.5681, 91.6845),
 (16, 'Banshbaria', 22.5492, 91.6909),
 (17, ' Chhoto Kumira', 22.5224, 91.70401),
 (18, 'Jaramtal', 22.4994, 91.71432),
 (19, 'Shitalpur', 22.4637, 91.7277),
 (20, 'Madam Bibir Hat', 22.43678, 91.7388),
 (21, 'Faujdarhat', 22.40171, 91.7569);

-- -----

--

-- Table structure for table `reason`

--

```
CREATE TABLE IF NOT EXISTS `reason` (
  `reason_id` int(11) NOT NULL AUTO_INCREMENT,
  `reason_name` varchar(30) NOT NULL,
  PRIMARY KEY (`reason_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```

--
-- Dumping data for table `reason`
--

INSERT INTO `reason` (`reason_id`, `reason_name`) VALUES
(1, 'Face To Face Attack '),
(2, 'Wrong Overtaking '),
(3, 'Careless Driving'),
(4, 'Others');

-----

--
-- Table structure for table `season`
--

CREATE TABLE IF NOT EXISTS `season` (
  `season_id` int(11) NOT NULL AUTO_INCREMENT,
  `season_name` varchar(30) NOT NULL,
  PRIMARY KEY (`season_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

--
-- Dumping data for table `season`
--

INSERT INTO `season` (`season_id`, `season_name`) VALUES
(1, 'Summer'),
(2, 'Rainy'),
(3, 'Winter');

```

```

-----

--
-- Table structure for table `vehicle_type`
--

CREATE TABLE IF NOT EXISTS `vehicle_type` (
  `type_id` int(11) NOT NULL AUTO_INCREMENT,
  `vehicle_name` varchar(30) NOT NULL,
  PRIMARY KEY (`type_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;

--
-- Dumping data for table `vehicle_type`
--

INSERT INTO `vehicle_type` (`type_id`, `vehicle_name`) VALUES
(1, 'Bus'),
(2, 'Truck'),
(3, 'Maxi'),
(4, 'Covered Van'),
(5, 'Pick Up'),
(6, 'CNG'),
(7, 'Private Car'),
(8, 'Micro Bus'),
(9, 'Bike'),
(10, 'Long Vehicle'),
(11, 'Jip'),
(12, 'Rickshaw'),
(13, 'Bicycle'),

```

(14, 'Leguna');

--

-- Constraints for dumped tables

--

--

-- Constraints for table `accident`

--

ALTER TABLE `accident`

ADD CONSTRAINT `accident\_ibfk\_1` FOREIGN KEY (`time\_id`) REFERENCES
`day\_time` (`time\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `accident\_ibfk\_2` FOREIGN KEY (`location\_id`) REFERENCES
`location` (`location\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `accident\_ibfk\_3` FOREIGN KEY (`type\_id`) REFERENCES
`vehicle\_type` (`type\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `accident\_ibfk\_4` FOREIGN KEY (`reason\_id`) REFERENCES
`reason` (`reason\_id`) ON DELETE CASCADE ON UPDATE CASCADE,

ADD CONSTRAINT `accident\_ibfk\_5` FOREIGN KEY (`season\_id`) REFERENCES
`season` (`season\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS
*/;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

/*

* To change this template, choose Tools | Templates

* and open the template in the editor.

*/

```
package roadaccident;
```

```
import java.awt.HeadlessException;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.SQLException;
```

```
public class Accident extends javax.swing.JFrame {
```

```
    String a_str1,a_str2,a_str3,a_str4,a_str5,a_str6,a_str7,a_str8,a_str9;
```

```
    /**
```

```
     * Creates new form Accident
```

```
     */
```

```
    public Accident() {
```

```
        initComponents();
```

```
        setDefaultCloseOperation(EXIT_ON_CLOSE);
```

```
    }
```

```
    /**
```

```
     * This method is called from within the constructor to initialize the form.
```

```
     * WARNING: Do NOT modify this code. The content of this method is always
```

```
     * regenerated by the Form Editor.
```

```
     */
```

```
    @SuppressWarnings("unchecked")
```

```
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```

private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel12 = new javax.swing.JLabel();
    a_bt2 = new javax.swing.JButton();
    a_bt3 = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();
    jPanel2 = new javax.swing.JPanel();
    a_tx2 = new javax.swing.JTextField();
    a_bt1 = new javax.swing.JButton();
    jLabel19 = new javax.swing.JLabel();
    a_tx8 = new javax.swing.JTextField();
    a_tx5 = new javax.swing.JTextField();
    jLabel14 = new javax.swing.JLabel();
    jLabel16 = new javax.swing.JLabel();
    jLabel21 = new javax.swing.JLabel();
    jButton2 = new javax.swing.JButton();
    jLabel20 = new javax.swing.JLabel();
    a_tx6 = new javax.swing.JTextField();
    jLabel18 = new javax.swing.JLabel();
    a_tx9 = new javax.swing.JTextField();
    a_tx3 = new javax.swing.JTextField();
    jLabel17 = new javax.swing.JLabel();
    a_tx4 = new javax.swing.JTextField();
    jLabel15 = new javax.swing.JLabel();
    a_tx7 = new javax.swing.JTextField();
    a_tx1 = new javax.swing.JTextField();
    jLabel22 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```
jLabel12.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N  
jLabel12.setText("TABLE NAME : ACCIDENT");
```

```
a_bt2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
a_bt2.setText("Back ");  
a_bt2.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        a_bt2ActionPerformed(evt);  
    }  
});
```

```
a_bt3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
a_bt3.setText("Back To Main Page");  
a_bt3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        a_bt3ActionPerformed(evt);  
    }  
});
```

```
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N  
jLabel1.setText("For Information Storing");
```

```
jButton1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
jButton1.setText("Next page");  
jButton1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton1ActionPerformed(evt);  
    }  
});
```

```

jPanel2.setBackground(new java.awt.Color(0, 153, 153));

a_bt1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
a_bt1.setText("STORE");
a_bt1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        a_bt1ActionPerformed(evt);
    }
});

jLabel19.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel19.setText("Reason_Id");

jLabel14.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel14.setText("Accident No");

jLabel16.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel16.setText("Time_Id");

jLabel21.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel21.setText("Injured");

jButton2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jButton2.setText("VIEW DATABASE TABLE");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jLabel20.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

```

```

jLabel20.setText("Season_Id");

jLabel18.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel18.setText("Type_Id");

jLabel17.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel17.setText("Location_Id");

jLabel15.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel15.setText("Date (yy-mm-dd)");

jLabel22.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel22.setText("Killed");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(113, 113, 113)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel19, javax.swing.GroupLayout.PREFERRED_SIZE,
113, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel14, javax.swing.GroupLayout.PREFERRED_SIZE,
172, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel16, javax.swing.GroupLayout.PREFERRED_SIZE,
134, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
    .addComponent(jLabel18,
        javax.swing.GroupLayout.Alignment.LEADING,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE)
    .addComponent(jLabel17,
        javax.swing.GroupLayout.Alignment.LEADING,
        javax.swing.GroupLayout.DEFAULT_SIZE, 126, Short.MAX_VALUE))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
    .addComponent(jLabel20,
        javax.swing.GroupLayout.Alignment.LEADING,
        javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)
    .addComponent(jLabel21,
        javax.swing.GroupLayout.Alignment.LEADING,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE)
    .addComponent(jLabel22,
        javax.swing.GroupLayout.Alignment.LEADING,
        javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE))
    .addComponent(jLabel15, javax.swing.GroupLayout.PREFERRED_SIZE,
        146, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(86, 86, 86)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
        .addComponent(a_bt1, javax.swing.GroupLayout.DEFAULT_SIZE, 137,
Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
222, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
        .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
        .addComponent(a_tx9, javax.swing.GroupLayout.DEFAULT_SIZE,
195, Short.MAX_VALUE)
        .addComponent(a_tx8)
        .addComponent(a_tx1)
        .addComponent(a_tx2)
        .addComponent(a_tx3)
        .addComponent(a_tx4)
        .addComponent(a_tx5)
        .addComponent(a_tx6)
        .addComponent(a_tx7))
        .addGap(0, 0, Short.MAX_VALUE))))
);
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(a_tx1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(13, 13, 13)
        .addComponent(a_tx2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(15, 15, 15))
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(jLabel14)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(jLabel15,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(7, 7, 7)))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(a_tx3)
    .addComponent(jLabel16, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel2Layout.createSequentialGroup()

        .addComponent(jLabel17,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel18,
javax.swing.GroupLayout.PREFERRED_SIZE, 24,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(jPanel2Layout.createSequentialGroup()

            .addComponent(a_tx4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(15, 15, 15)

            .addComponent(a_tx5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGap(10, 10, Short.MAX_VALUE)

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel19, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 20,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(a_tx6, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(a_tx7, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel20, javax.swing.GroupLayout.PREFERRED_SIZE,
28, javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(a_tx8, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel21, javax.swing.GroupLayout.PREFERRED_SIZE,
24, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel22, javax.swing.GroupLayout.PREFERRED_SIZE,
17, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(a_tx9, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
39, Short.MAX_VALUE)

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
    .addComponent(jButton2)
    .addComponent(a_bt1))
.addContainerGap()
);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addGap(0, 73, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(21, 21, 21))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addComponent(a_bt2, javax.swing.GroupLayout.PREFERRED_SIZE,
173, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(89, 89, 89)

```

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(a_bt3, javax.swing.GroupLayout.PREFERRED_SIZE,
222, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(184, 184, 184))))
.addGroup(jPanel1Layout.createSequentialGroup()
    .addGap(35, 35, 35)
    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 256,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap())
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE,
308, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(179, 179, 179))
);
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(25, 25, 25)
        .addComponent(jLabel1)

```

```

        .addGap(18, 18, 18)
        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(jButton1)
        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
            .addComponent(a_bt2)
            .addComponent(a_bt3))
        .addContainerGap(17, Short.MAX_VALUE)
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(27, 27, 27)

```

```

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addContainerGap()
    );

    pack();
} // </editor-fold>

```

```

private void a_bt2ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    setVisible(false);
    Admin1 a=new Admin1();
    a.setVisible(true);

```

```

// TODO add your handling code here:
}

```

```

private void a_bt3ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    setVisible(false);
    Login l=new Login();
    l.setVisible(true);
}

```

```

private void a_bt1ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    a_str1=a_tx1.getText();
    a_str2=a_tx2.getText();
    a_str3=a_tx3.getText();
    a_str4=a_tx4.getText();

```

```

a_str5=a_tx5.getText();
a_str6=a_tx6.getText();
a_str7=a_tx7.getText();
a_str8=a_tx8.getText();
a_str9=a_tx9.getText();

// TODO add your handling code here:
try{
String url="jdbc:mysql://localhost:3306/road_accident";
String user="root";
String password="root";
Connection conn=(Connection)DriverManager.getConnection(url, user, password);
String query="insert into
accident(accident_no,date,time_id,location_id,type_id,reason_id,season_id,injured,killed
)"+
values('"+a_str1+"','"+a_str2+"','"+a_str3+"','"+a_str4+"','"+a_str5+"','"+a_str6+"','"+a_str7
+"','"+a_str8+"','"+a_str9+"')";
    PreparedStatement prepareStm=conn.prepareStatement(query);
    prepareStm.execute();

a_tx1.setText("");
a_tx2.setText("");
a_tx3.setText("");
a_tx4.setText("");
a_tx5.setText("");
a_tx6.setText("");
a_tx7.setText("");
a_tx8.setText("");
a_tx9.setText("");

conn.close();

```

```

    }
    catch(SQLException|HeadlessException ex)
    {
        System.out.println(ex);

    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    setVisible(false);
    Accident2 ac2=new Accident2();
    ac2.setVisible(true);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    Accident_Table at=new Accident_Table();
    at.setVisible(true);

}

/**
 * @param args the command line arguments

```

```

*/
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Accident.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Accident.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Accident.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```
java.util.logging.Logger.getLogger(Accident.class.getName()).log(java.util.logging.Level.  
SEVERE, null, ex);
```

```
}
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
    public void run() {
```

```
        new Accident().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton a_bt1;
```

```
private javax.swing.JButton a_bt2;
```

```
private javax.swing.JButton a_bt3;
```

```
private javax.swing.JTextField a_tx1;
```

```
private javax.swing.JTextField a_tx2;
```

```
private javax.swing.JTextField a_tx3;
```

```
private javax.swing.JTextField a_tx4;
```

```
private javax.swing.JTextField a_tx5;
```

```
private javax.swing.JTextField a_tx6;
```

```
private javax.swing.JTextField a_tx7;
```

```
private javax.swing.JTextField a_tx8;
```

```
private javax.swing.JTextField a_tx9;
```

```
private javax.swing.JButton jButton1;
```

```
private javax.swing.JButton jButton2;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel12;
```

```
private javax.swing.JLabel jLabel14;
```

```

private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
// End of variables declaration
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadaccident;

import java.awt.Color;
import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class Accident2 extends javax.swing.JFrame {

```

```

/**
 * Creates new form Accident2
 */
public Accident2() {
    initComponents();
    getContentPane().setBackground(new Color(204,204,255));
    Fillcombo();
}

private void Fillcombo()
{
    try
    {
        String url="jdbc:mysql://localhost:3306/road_accident";
        String user="root";
        String pass="root";
        Connection conn=(Connection)DriverManager.getConnection(url,user,pass);
        String query= "select * from accident";
        PreparedStatement preparedStmt = conn.prepareStatement(query);
        preparedStmt.execute();
        ResultSet rs1 = preparedStmt.getResultSet();

        while(rs1.next())
        {
            int ac_no=rs1.getInt("accident_no");
            ac2_com1.addItem(ac_no);
            ac2_com2.addItem(ac_no);
            ac2_com3.addItem(ac_no);
        }
    }
    catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}

```

```

        ac2_com4.addItem(ac_no);

    }

    conn.close();
}
catch(SQLException | HeadlessException e)
{
    System.out.println(e);
}
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    ac_bt9 = new javax.swing.JButton();
    ac_bt10 = new javax.swing.JButton();
    jPanel1 = new javax.swing.JPanel();

```

```
ac_bt2 = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();
ac2_com1 = new javax.swing.JComboBox();
ac_bt1 = new javax.swing.JButton();
jPanel2 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();
ac_tx1 = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
ac2_com2 = new javax.swing.JComboBox();
jLabel5 = new javax.swing.JLabel();
ac_bt3 = new javax.swing.JButton();
ac_bt4 = new javax.swing.JButton();
jPanel3 = new javax.swing.JPanel();
ac_bt6 = new javax.swing.JButton();
jLabel10 = new javax.swing.JLabel();
ac_tx2 = new javax.swing.JTextField();
jLabel9 = new javax.swing.JLabel();
ac2_com3 = new javax.swing.JComboBox();
jLabel4 = new javax.swing.JLabel();
ac_bt5 = new javax.swing.JButton();
jPanel4 = new javax.swing.JPanel();
ac_tx3 = new javax.swing.JTextField();
jLabel11 = new javax.swing.JLabel();
ac_bt8 = new javax.swing.JButton();
ac_bt7 = new javax.swing.JButton();
jLabel8 = new javax.swing.JLabel();
ac2_com4 = new javax.swing.JComboBox();
jLabel13 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel1.setText("For Information Removing");
```

```
jLabel3.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel3.setText("For Information Updating");
```

```
ac_bt9.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
ac_bt9.setText("BACK");
ac_bt9.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ac_bt9ActionPerformed(evt);
    }
});
```

```
ac_bt10.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
ac_bt10.setText("BACK TO MAIN PAGE");
ac_bt10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ac_bt10ActionPerformed(evt);
    }
});
```

```
jPanel1.setBackground(new java.awt.Color(204, 255, 204));
```

```
ac_bt2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
ac_bt2.setText("VIEW DATABASE TABLE");
ac_bt2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ac_bt2ActionPerformed(evt);
    }
});
```

```

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel2.setText("Select Accident No");

ac2_com1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
ac2_com1.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Select Accident No" }));

ac_bt1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
ac_bt1.setText("REMOVE");
ac_bt1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ac_bt1ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(66, 66, 66)
        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(42, 42, 42)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(ac2_com1, javax.swing.GroupLayout.PREFERRED_SIZE,
233, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(ac_bt1)
            .addGap(27, 27, 27)
            .addComponent(ac_bt2, javax.swing.GroupLayout.PREFERRED_SIZE,
220, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(63, Short.MAX_VALUE)
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
        .addComponent(jLabel2)
        .addComponent(ac2_com1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
        .addComponent(ac_bt1)
        .addComponent(ac_bt2))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

jPanel2.setBackground(new java.awt.Color(102, 255, 204));

```

```
jLabel6.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
jLabel6.setText("New Date");
```

```
jLabel7.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
jLabel7.setText("Updating Date Column");
```

```
ac2_com2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
ac2_com2.setModel(new javax.swing.DefaultComboBoxModel(new String[] {  
"Select Accident No" }));
```

```
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
jLabel5.setText("Select Accident No");
```

```
ac_bt3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
ac_bt3.setText("UPDATE");
```

```
ac_bt3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        ac_bt3ActionPerformed(evt);  
    }  
});
```

```
ac_bt4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
ac_bt4.setText("VIEW DATABASE TABLE");
```

```
ac_bt4.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        ac_bt4ActionPerformed(evt);  
    }  
});
```

```
javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
```

```
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
```

```
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(67, 67, 67)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(jLabel5,

javax.swing.GroupLayout.PREFERRED_SIZE, 172,

javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabel6,

javax.swing.GroupLayout.PREFERRED_SIZE, 113,

javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(28, 28, 28)
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel2Layout.createSequentialGroup()

        .addComponent(ac_bt3)

        .addGap(24, 24, 24)

        .addComponent(ac_bt4,

javax.swing.GroupLayout.PREFERRED_SIZE, 224,

javax.swing.GroupLayout.PREFERRED_SIZE))
```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

        .addComponent(ac_tx1,
javax.swing.GroupLayout.PREFERRED_SIZE, 230,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(ac2_com2,
javax.swing.GroupLayout.PREFERRED_SIZE, 230,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE,
203, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(60, Short.MAX_VALUE)
    );
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(jLabel7)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel5)
        .addComponent(ac2_com2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jLabel6)
        .addComponent(ac_tx1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
        .addComponent(ac_bt3)
        .addComponent(ac_bt4))
.addContainerGap()
);

jPanel3.setBackground(new java.awt.Color(0, 204, 204));

ac_bt6.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
ac_bt6.setText("VIEW DATABASE TABLE");
ac_bt6.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ac_bt6ActionPerformed(evt);
    }
});

jLabel10.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel10.setText("New Injured Number");

jLabel9.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel9.setText("Select Accident No");

ac2_com3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

```

```

        ac2_com3.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Select Accident No" }));

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel4.setText("Updating Injured Column");

ac_bt5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
ac_bt5.setText("UPDATE");
ac_bt5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ac_bt5ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .add(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addContainerGap()

.addGroup(jPanel3Layout.createSequentialGroup()
                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addGroup(jPanel3Layout.createSequentialGroup()
                            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .add(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE,
241, javax.swing.GroupLayout.PREFERRED_SIZE)
                                .addGap(0, 0, Short.MAX_VALUE))
                            .addGroup(jPanel3Layout.createSequentialGroup()
                                .addGroup(jPanel3Layout.createSequentialGroup()
                                    .addGap(53, 53, 53)

```

```

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel9,
javax.swing.GroupLayout.PREFERRED_SIZE, 160,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel10,
javax.swing.GroupLayout.PREFERRED_SIZE, 178,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(38, 38, 38)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel3Layout.createSequentialGroup()
        .addComponent(ac_bt5,
javax.swing.GroupLayout.PREFERRED_SIZE, 104,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(ac_bt6,
javax.swing.GroupLayout.PREFERRED_SIZE, 227,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(ac2_com3,
javax.swing.GroupLayout.PREFERRED_SIZE, 232,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(ac_tx2,
javax.swing.GroupLayout.PREFERRED_SIZE, 234,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(54, Short.MAX_VALUE))))
);
jPanel3Layout.setVerticalGroup(

```

```

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel4)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
            .addComponent(jLabel9)
            .addComponent(ac2_com3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
            .addComponent(jLabel10)
            .addComponent(ac_tx2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
            .addComponent(ac_bt5)
            .addComponent(ac_bt6))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

```

```
jPanel4.setBackground(new java.awt.Color(102, 204, 255));
```

```
jLabel11.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
jLabel11.setText("Select Accident No");
```

```
ac_bt8.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
ac_bt8.setText("VIEW DATABASE TABLE");
```

```
ac_bt8.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        ac_bt8ActionPerformed(evt);  
    }  
});
```

```
ac_bt7.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
ac_bt7.setText("UPDATE");
```

```
ac_bt7.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        ac_bt7ActionPerformed(evt);  
    }  
});
```

```
jLabel8.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
jLabel8.setText("Updating Killed Column");
```

```
ac2_com4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
ac2_com4.setModel(new javax.swing.DefaultComboBoxModel(new String[] {  
"Select Accident No" }));
```

```
jLabel13.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
```

```
jLabel13.setText("New Killed Number");
```

```

    javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
    jPanel4.setLayout(jPanel4Layout);
    jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE,
276, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jPanel4Layout.createSequentialGroup()
                .addGap(57, 57, 57)

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addComponent(jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 156,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel13,
javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE))

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel4Layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addGroup(jPanel4Layout.createSequentialGroup()

```

```

        .addGap(56, 56, 56)
        .addComponent(ac_bt7,
javax.swing.GroupLayout.PREFERRED_SIZE, 111,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(ac_bt8,
javax.swing.GroupLayout.PREFERRED_SIZE, 237,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel4Layout.createSequentialGroup())
        .addGap(60, 60, 60)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(ac_tx3,
javax.swing.GroupLayout.PREFERRED_SIZE, 240,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(ac2_com4,
javax.swing.GroupLayout.PREFERRED_SIZE, 240,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(37, Short.MAX_VALUE))))
    );
    jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addComponent(jLabel8)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(jLabel11)
        .addComponent(ac2_com4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
        .addComponent(ac_tx3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel13))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addComponent(ac_bt8)
        .addComponent(ac_bt7))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

```

```

        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(187, 187, 187)
            .addComponent(ac_bt9,
javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(ac_bt10,
javax.swing.GroupLayout.PREFERRED_SIZE, 194,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(35, 35, 35)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 262,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 262,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGap(0, 35, Short.MAX_VALUE))
    .addGroup(layout.createSequentialGroup()
        .addGap(35, 35, 35)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 4,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(4, 4, 4)
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel3)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addGroup(layout.createSequentialGroup()
            .addGap(288, 288, 288)
            .addComponent(jLabel12)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(18, 18, 18)
                .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 21,
Short.MAX_VALUE)
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)))
    );

    pack();
} // </editor-fold>

private void ac_bt2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

```

```

        setVisible(false);
        Accident_Table2 act2=new Accident_Table2();
        act2.setVisible(true);

    }

    private void ac_bt4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        Accident_Table2 act2=new Accident_Table2();
        act2.setVisible(true);

    }

    private void ac_bt6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        Accident_Table2 act2=new Accident_Table2();
        act2.setVisible(true);

    }

    private void ac_bt8ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        setVisible(false);
        Accident_Table2 act2=new Accident_Table2();
        act2.setVisible(true);

    }

```

```

private void ac_bt9ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setVisible(false);
    Accident ac=new Accident();
    ac.setVisible(true);

}

```

```

private void ac_bt10ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    setVisible(false);
    Login l=new Login();
    l.setVisible(true);

}

```

```

private void ac_bt1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int d=JOptionPane.showConfirmDialog(null, "Do You Really Want To Remove This
Item ","Remove",JOptionPane.YES_NO_OPTION);

    if(d==0)
    try{

        String acc_no=ac2_com1.getSelectedItemAt(0).toString();

        String url="jdbc:mysql://localhost:3306/road_accident";
        String user="root";

```

```

String password="root";
Connection conn=(Connection)DriverManager.getConnection(url, user, password);
String query=" delete from accident where accident_no='"+acc_no+"' ";
PreparedStatement preparedStm=conn.prepareStatement(query);
preparedStm.execute();
ac2_com1.setSelectedIndex(0);

//Fillcombo();
conn.close();

}
catch(SQLException|HeadlessException e)
{

System.out.println(""+e);
}

}

private void ac_bt5ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

String ac_str3,ac_str4;

ac_str3=ac_tx2.getText();

```

```

ac_str4=ac2_com3.getSelectedItemId().toString();

try{

    String url="jdbc:mysql://localhost:3306/road_accident";
    String user="root";
    String password="root";
    Connection conn=(Connection)DriverManager.getConnection(url, user, password);
    String query="update accident set injured='"+ac_str3+"' where
accident_no='"+ac_str4+"'";
    PreparedStatement preparedStm=conn.prepareStatement(query);
    preparedStm.execute();

    ac_tx2.setText("");
    ac2_com3.setSelectedIndex(0);

    conn.close();

}

catch(SQLException|HeadlessException e)
{
    System.out.println(e);

}

}

```

```

private void ac_bt3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    String ac_str1,ac_str2;

    ac_str1=ac_tx1.getText();

    ac_str2=ac2_com2.getSelectedItem().toString();

    try{

        String url="jdbc:mysql://localhost:3306/road_accident";
        String user="root";
        String password="root";
        Connection conn=(Connection)DriverManager.getConnection(url, user, password);
        String query="update accident set date='"+ac_str1+"' where
accident_no='"+ac_str2+"'";
        PreparedStatement preparedStm=conn.prepareStatement(query);
        preparedStm.execute();

        ac_tx1.setText("");
        ac2_com2.setSelectedIndex(0);

        conn.close();

    }

    catch(SQLException|HeadlessException e)
    {
        System.out.println(e);
    }
}

```

```
}
```

```
}
```

```
private void ac_bt7ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:
```

```
        String ac_str5,ac_str6;
```

```
        ac_str5=ac_tx3.getText();
```

```
        ac_str6=ac2_com4.getSelectedItem().toString();
```

```
        try{
```

```
            String url="jdbc:mysql://localhost:3306/road_accident";
```

```
            String user="root";
```

```
            String password="root";
```

```
            Connection conn=(Connection)DriverManager.getConnection(url, user, password);
```

```
            String query="update accident set killed='"+ac_str5+"' where
```

```
accident_no='"+ac_str6+"'";
```

```
            PreparedStatement preparedStm=conn.prepareStatement(query);
```

```
            preparedStm.execute();
```

```
            ac_tx3.setText("");
```

```
            ac2_com4.setSelectedIndex(0);
```

```
            conn.close();
```

```
        }
```

```

        catch(SQLException|HeadlessException e)
        {
            System.out.println(e);

        }

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
        and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        }
    }

```

```

    }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Accident2.class.getName()).log(java.util.logging.Level
I.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Accident2.class.getName()).log(java.util.logging.Level
I.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Accident2.class.getName()).log(java.util.logging.Level
I.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Accident2.class.getName()).log(java.util.logging.Level
I.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Accident2().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JComboBox ac2_com1;
private javax.swing.JComboBox ac2_com2;
private javax.swing.JComboBox ac2_com3;

```

```
private javax.swing.JComboBox ac2_com4;  
private javax.swing.JButton ac_bt1;  
private javax.swing.JButton ac_bt10;  
private javax.swing.JButton ac_bt2;  
private javax.swing.JButton ac_bt3;  
private javax.swing.JButton ac_bt4;  
private javax.swing.JButton ac_bt5;  
private javax.swing.JButton ac_bt6;  
private javax.swing.JButton ac_bt7;  
private javax.swing.JButton ac_bt8;  
private javax.swing.JButton ac_bt9;  
private javax.swing.JTextField ac_tx1;  
private javax.swing.JTextField ac_tx2;  
private javax.swing.JTextField ac_tx3;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel10;  
private javax.swing.JLabel jLabel11;  
private javax.swing.JLabel jLabel12;  
private javax.swing.JLabel jLabel13;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JPanel jPanel4;
```

```

    // End of variables declaration
}
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadaccident;

import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Calendar;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.LinkedList;
import java.util.Map;

public class Accident_Prone_Vehicle extends javax.swing.JFrame {

    /**
     * Creates new form Accident_Prone_Vehicle
     */
    public Accident_Prone_Vehicle() {
        initComponents();
    }

```

```

        int i,m,n;
        double f[]= new double[100];
        // Arrays.fill(f, 0);
        Calendar now = Calendar.getInstance();

        int start_year=2007;
        int current_year = now.get(Calendar.YEAR);

try{

        String url="jdbc:mysql://localhost:3306/road_accident";
        String user="root";
        String pass="root";
        Connection conn=(Connection)DriverManager.getConnection(url, user, pass);

        String query="Select count(type_id) as total_vehicle from vehicle_type";
        PreparedStatement prepareStm=conn.prepareStatement(query);
        prepareStm.execute();
        ResultSet rs2=prepareStm.getResultSet();

        int no_of_vehicles = 0;
        while(rs2.next())
        {
            no_of_vehicles = rs2.getInt("total_vehicle");
        }

```

```

Map<String, Double> unsortMap = new HashMap<String, Double>();

String vehicle[]=new String[100];

double temp,total_percentage =0;

String cur_vehicle="";

for(int j = 1; j<=no_of_vehicles;j++ )
{
    temp = 0;
    for(i=(current_year-1);i>=start_year;i--)
    {

        m = 0;
        n=0;
        String query1="Select a.vehicle_name,count(b.accident_no) as
total_accident,sum(b.killed) as total_killed from vehicle_type a left join accident b on
a.type_id=b.type_id where b.type_id='"+j+"' and date like '"+i+"%' GROUP BY
a.vehicle_name ";
        prepareStm=conn.prepareStatement(query1);
        prepareStm.execute();
        rs2=prepareStm.getResultSet();

        while(rs2.next())
        {
            m=rs2.getInt("total_accident");

```

```

        n=rs2.getInt("total_killed");
        //System.out.println(""+n);
        cur_vehicle = rs2.getString("a.vehicle_name");
    }

    temp=temp+m*n;
    //System.out.println(""+temp);
}

    vehicle[j] = cur_vehicle;
    f[j] = temp;

    total_percentage += f[j];

}

for(i = 1; i<= no_of_vehicles; i++)
{
    temp = (f[i]*100)/total_percentage;
    //temp = temp * 100;
    //System.out.println(" "+temp);

    unsortMap.put(vehicle[i],temp);
}
Map<String, Double> sortedMap = sortByValue(unsortMap);
printMap(sortedMap);

}
catch(SQLException|HeadlessException e)

```

```

{
    System.out.println(e);
}

```

```

}

```

```

private static Map<String, Double> sortByValue(Map<String,Double > unsortMap) {

```

```

    LinkedList<Map.Entry<String, Double>> list;
    list = new LinkedList<Map.Entry<String, Double>>(unsortMap.entrySet());
    Collections.sort(list, new Comparator<Map.Entry<String, Double>>() {
        public int compare(Map.Entry<String, Double> o1,
                           Map.Entry<String, Double> o2) {
            return (o2.getValue()).compareTo(o1.getValue());
        }
    });
    Map<String, Double> sortedMap = new LinkedHashMap<String, Double>();
    for (Map.Entry<String, Double> entry : list) {
        sortedMap.put(entry.getKey(), entry.getValue());
    }
    return sortedMap;
}

```

```

public static <K, V> void printMap(Map<K, V> map)

```

```

{
    for (Map.Entry<K, V> entry : map.entrySet())
    {

```

```

        System.out.println("vehicle : " + entry.getKey()
+ " Value : " + entry.getValue());
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 702, Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 510, Short.MAX_VALUE)
    );

    pack();
} // </editor-fold>

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)
">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Accident_Prone_Vehicle.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Accident_Prone_Vehicle.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(Accident_Prone_Vehicle.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(Accident_Prone_Vehicle.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Accident_Prone_Vehicle().setVisible(true);
    }
});
}
// Variables declaration - do not modify
// End of variables declaration
}
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package roadaccident;

```

```

import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

```

```

import java.sql.SQLException;
import java.util.Calendar;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.LinkedList;
import java.util.Map;

import javax.swing.JFrame;

import org.jfree.ui.RefineryUtilities;

public class Accident_Prone_Vehicle1 {

    public Accident_Prone_Vehicle1() {
        int i,m,n;
        double f[]= new double[100];
        // Arrays.fill(f, 0);
        Calendar now = Calendar.getInstance();

        int start_year=2007;
        int current_year = now.get(Calendar.YEAR);

        try{

            String url="jdbc:mysql://localhost:3306/road_accident";
            String user="root";

```

```
String pass="root";
Connection conn=(Connection)DriverManager.getConnection(url, user, pass);
```

```
String query="Select count(type_id) as total_vehicle from vehicle_type";
PreparedStatement prepareStm=conn.prepareStatement(query);
prepareStm.execute();
ResultSet rs2=prepareStm.getResultSet();
```

```
int no_of_vehicles = 0;
while(rs2.next())
{
    no_of_vehicles = rs2.getInt("total_vehicle");
}
```

```
Map<String, Double> unsortMap = new HashMap<String, Double>();
```

```
String vehicle[]=new String[100];
```

```
double temp,total_percentage =0;
```

```
String cur_vehicle="";
```

```
for(int j = 1; j<=no_of_vehicles;j++ )
{
    temp = 0;
    for(i=(current_year-1);i>=start_year;i--)
    {
```

```

        m = 0;
        n=0;
        String query1="Select a.vehicle_name,count(b.accident_no) as
total_accident,sum(b.killed) as total_killed from vehicle_type a left join accident b on
a.type_id=b.type_id where b.type_id='"+j+"' and date like '"+i+"%' GROUP BY
a.vehicle_name ";
        prepareStm=conn.prepareStatement(query1);
        prepareStm.execute();
        rs2=prepareStm.getResultSet();

        while(rs2.next())
        {
            m=rs2.getInt("total_accident");

            n=rs2.getInt("total_killed");
            //System.out.println(""+n);
            cur_vehicle = rs2.getString("a.vehicle_name");
        }

        temp=temp+m*n;
        //

        CreateChart cc = new CreateChart("Accident Prone Vehicles", "Road Accident",
sortedMap);
        cc.pack();
        cc.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        //cc.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        RefineryUtilities.centerFrameOnScreen(cc);
        cc.setVisible(true);

```

```

    }
    catch(SQLException|HeadlessException e)
    {
        System.out.println(e);
    }
}

```

```

private static Map<String, Double> sortByValue(Map<String,Double > unsortMap) {

```

```

    LinkedList<Map.Entry<String, Double>> list;
    list = new LinkedList<Map.Entry<String, Double>>(unsortMap.entrySet());
    Collections.sort(list, new Comparator<Map.Entry<String, Double>>() {
        public int compare(Map.Entry<String, Double> o1,
            Map.Entry<String, Double> o2) {
            return (o2.getValue()).compareTo(o1.getValue());
        }
    });
    Map<String, Double> sortedMap = new LinkedHashMap<String, Double>();
    for (Map.Entry<String, Double> entry : list) {
        sortedMap.put(entry.getKey(), entry.getValue());
    }
    return sortedMap;
}

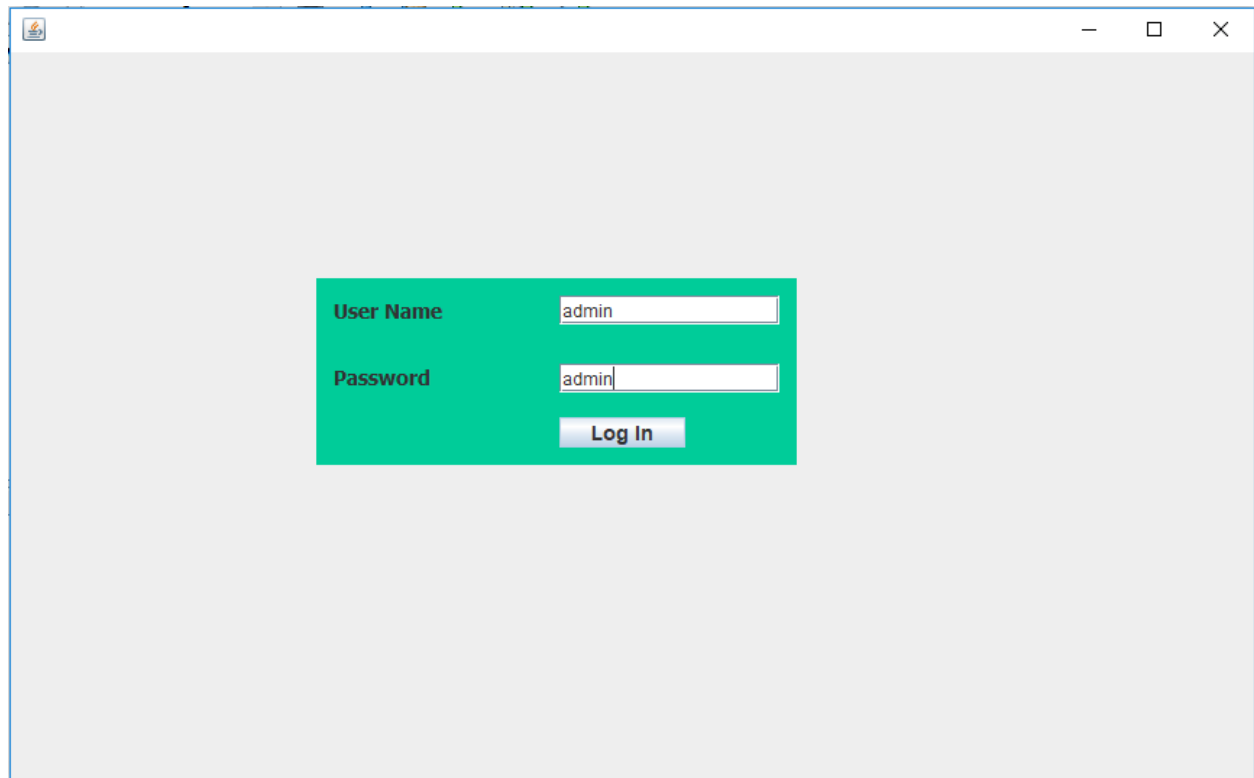
```

```
public static <K, V> void printMap(Map<K, V> map)

{
    for (Map.Entry<K, V> entry : map.entrySet())
    {

//          System.out.println("vehicle : " + entry.getKey()
//          + " Value : " + entry.getValue());
    }
}
```

B. SCREEN SHOTS



A screenshot of a web application window titled "Admin Login". The window has a light gray background and a standard Windows-style title bar with a minimize button, a maximize button, and a close button. In the center of the window is a green rectangular login form. The form contains two text input fields: the first is labeled "User Name" and contains the text "admin"; the second is labeled "Password" and contains the text "admin". Below these fields is a blue button with the text "Log In".

User Name	admin
Password	admin
Log In	

Fig.1.Admin Login



Fig.2. Select Table

TABLE NAME:VEHICLE TYPE

For Storing Information

Type Id

vehicle_name

STORE

View Database Table

For Removing Information

Select Vehicle Name

Select Vehicle Name

Remove

View Database Table

For Updating Information

Select Vehicle Name

Select Vehicle Name

New Vehicle Name

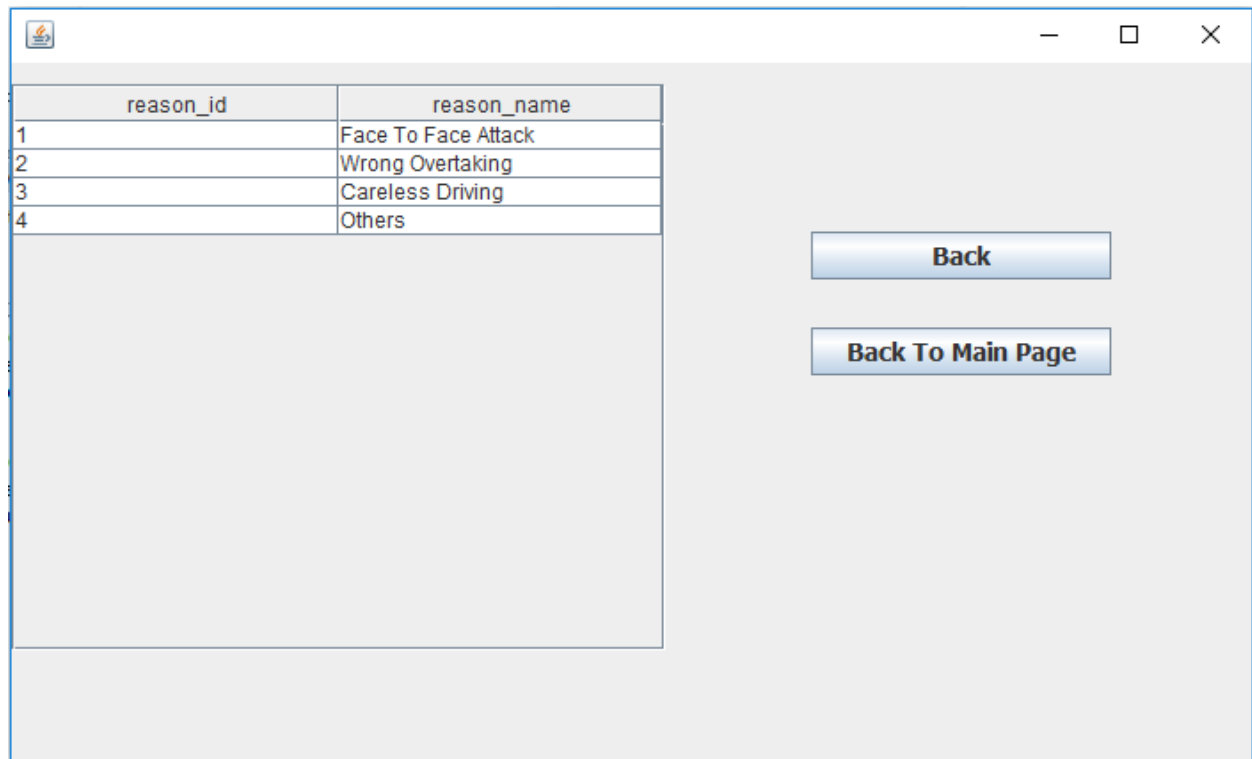
Update

View Database Table

Back

Back To Main Page

Fig.3.Vehicle Type Table



reason_id	reason_name
1	Face To Face Attack
2	Wrong Overtaking
3	Careless Driving
4	Others

Back

Back To Main Page

Fig.4. ACCIDENT REASON TABLE

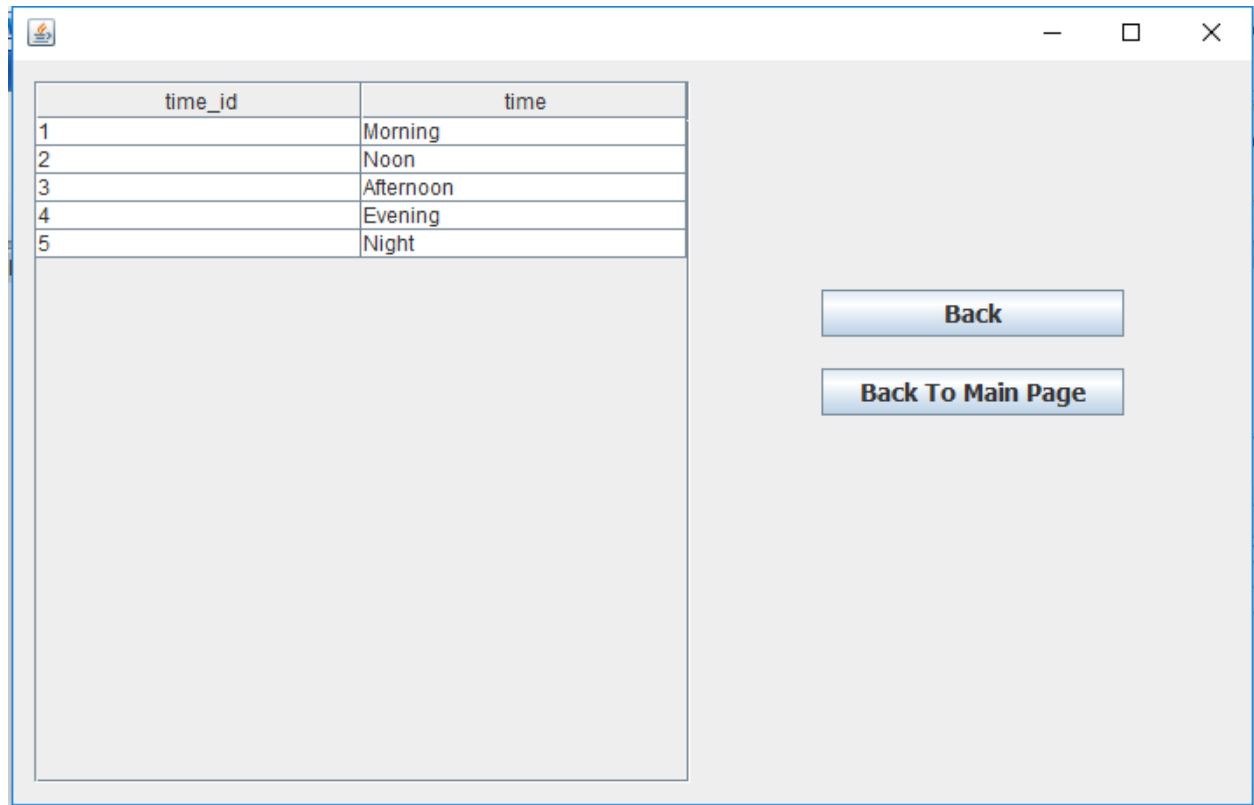
Table Name :Season

season_id	season_name
1	Summer
2	Rainy
3	Winter

Back

Back To Main Page

Fig.5. ACCIDENT SEASON TABLE



The image shows a web application window with a table and two buttons. The table has two columns: 'time_id' and 'time'. It contains five rows of data. Below the table is a large empty rectangular area. To the right of this area are two buttons: 'Back' and 'Back To Main Page'.

time_id	time
1	Morning
2	Noon
3	Afternoon
4	Evening
5	Night

Back

Back To Main Page

Fig.6. TIME OF THE DAY TABLE

location_id	location_name	lat	lon
1	Mohipal	23.009	91.382
2	Lalpol	22.983	91.412
3	East Chilonia	22.9668	91.44577
4	Lemua	22.95954	91.43957
5	Koska	22.9492	91.46403
6	Fazilpur	22.9548	91.48661
7	Muhurigonj	22.94498	91.5102
8	Braraiyarhat	22.8955	91.5325
9	Mithachara	22.8048	91.557
10	Mirsharai	22.77396	91.57297
11	Noyduaria	22.7374	91.5971
12	Nizampur	22.718	91.6056
13	Borodarogarhat	22.6778	91.62616
14	Panthasala	22.6372	91.6476
15	Barabkunda	22.5681	91.6845
16	Banshbaria	22.5492	91.6909
17	Chhoto Kumira	22.5224	91.70401
18	Jaramtal	22.4994	91.71432
19	Shitalpur	22.4637	91.7277
20	Madam Bibir Hat	22.43678	91.7388
21	Faujdarhat	22.40171	91.7569

Back

Back To Main Page

Fig.7. ACCIDENT LOCATION TABLE

TABLE NAME : ACCIDENT

For Information Storing

Accident No	
Date (yy-mm-dd)	
Time_Id	
Location_Id	
Type_Id	
Reason_Id	
Season_Id	
Injured	
Killed	

Fig.8. ACCIDENT TABLE

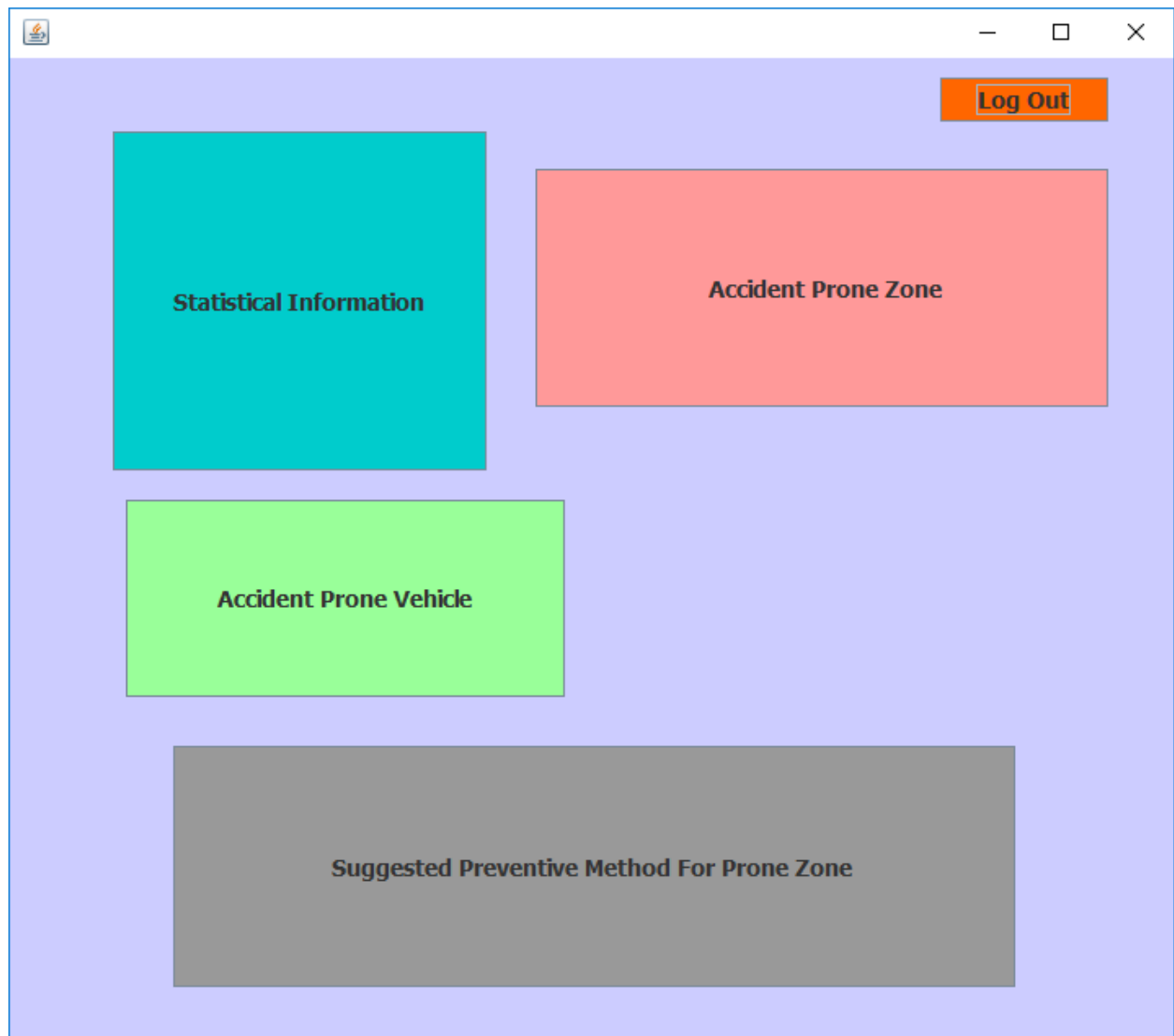


Fig.9. USER LOGIN

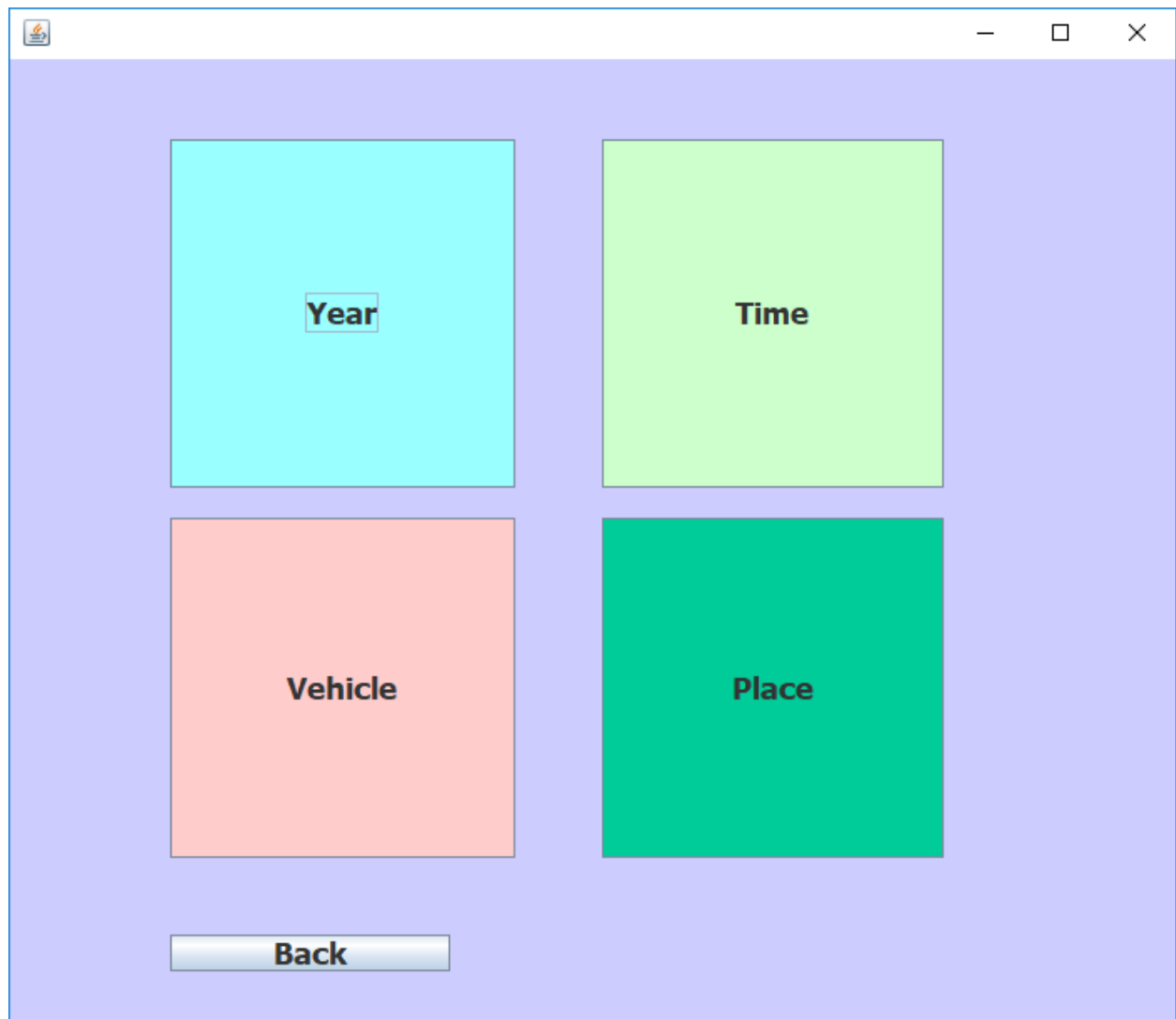


Fig.10. STATITICAL INFORMATION

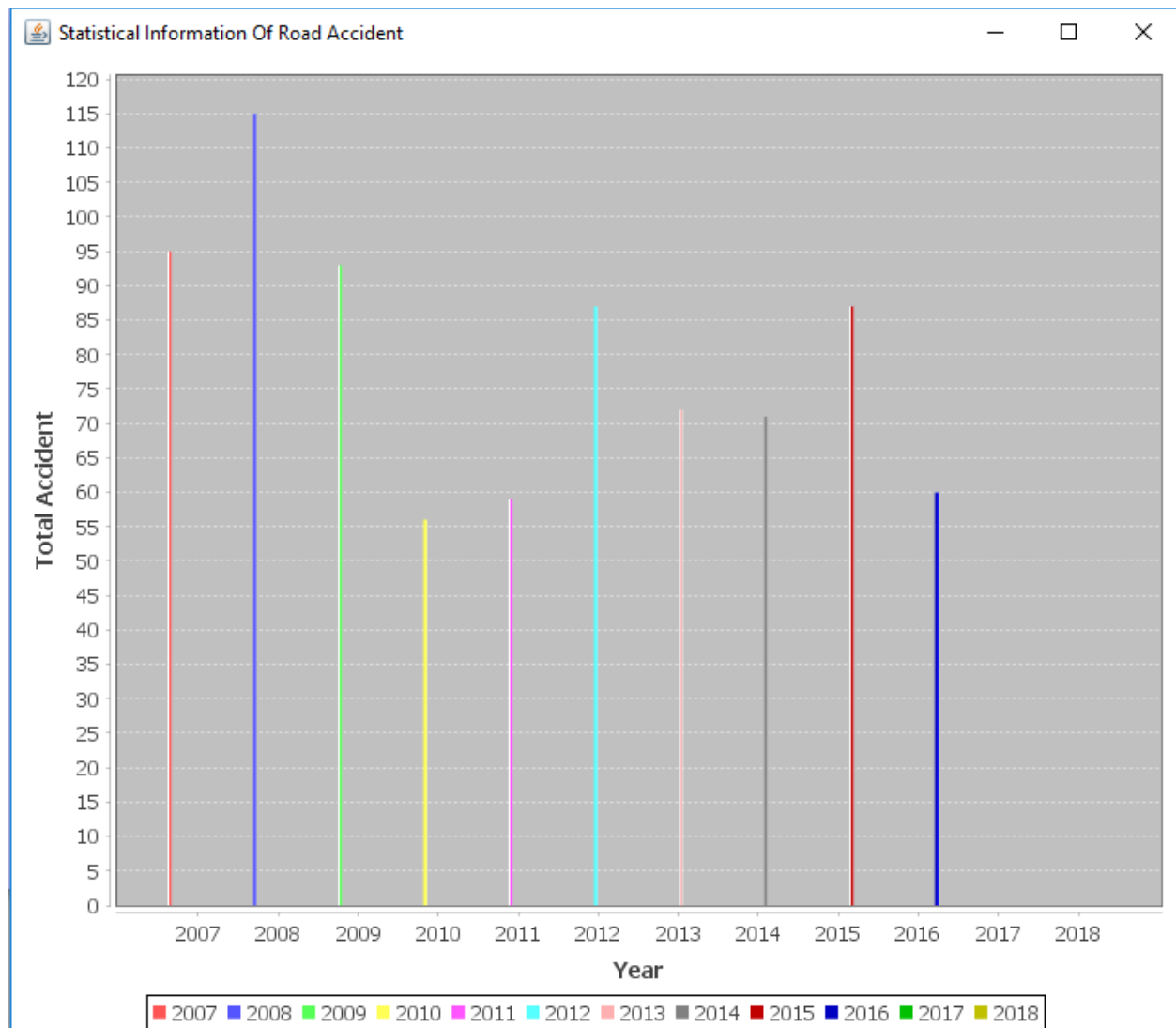


Fig.11.YEAR

Year	Time Of The Day	Total Number Of Accident
2007	Morning	17
2008	Morning	22
2009	Morning	9
2010	Morning	10
2011	Morning	11
2012	Morning	12
2013	Morning	12
2014	Morning	16
2015	Morning	21
2016	Morning	12
2007	Noon	15
2008	Noon	29
2009	Noon	16
2010	Noon	8
2011	Noon	6
2012	Noon	15
2013	Noon	11
2014	Noon	19
2015	Noon	9
2016	Noon	17
2007	Afternoon	20
2008	Afternoon	17
2009	Afternoon	17
2010	Afternoon	8
2011	Afternoon	10
2012	Afternoon	19
2013	Afternoon	11
2014	Afternoon	10
2015	Afternoon	15
2016	Afternoon	9
2007	Evening	17

Fig.12. TIME

Year	Vehicle Name	Number Of Total Accident
2007	Bus	17
2008	Bus	13
2009	Bus	10
2010	Bus	6
2011	Bus	10
2012	Bus	25
2013	Bus	17
2014	Bus	17
2015	Bus	20
2016	Bus	12
2007	Truck	20
2008	Truck	23
2009	Truck	14
2010	Truck	8
2011	Truck	21
2012	Truck	30
2013	Truck	23
2014	Truck	16
2015	Truck	25
2016	Truck	20
2007	Maxi	4
2008	Maxi	9
2009	Maxi	1
2010	Maxi	2
2011	Maxi	3
2012	Maxi	7
2013	Maxi	8
2014	Maxi	10

Fig.13. VEHICLE

Year	Location	Total Number Of Accident
2007	Mohipal	9
2008	Mohipal	6
2009	Mohipal	5
2010	Mohipal	5
2011	Mohipal	3
2012	Mohipal	16
2013	Mohipal	6
2014	Mohipal	7
2015	Mohipal	11
2016	Mohipal	2
2007	Lalpol	2
2008	Lalpol	3
2010	Lalpol	3
2011	Lalpol	3
2012	Lalpol	9
2013	Lalpol	6
2014	Lalpol	5
2015	Lalpol	8
2016	Lalpol	5
2008	East Chilonia	2
2009	East Chilonia	3
2011	East Chilonia	5
2012	East Chilonia	1
2013	East Chilonia	8
2014	East Chilonia	4
2015	East Chilonia	11
2007	Lemua	2
2008	Lemua	1
2009	Lemua	2
2010	Lemua	3
2011	Lemua	2

Fig.14. PLACE

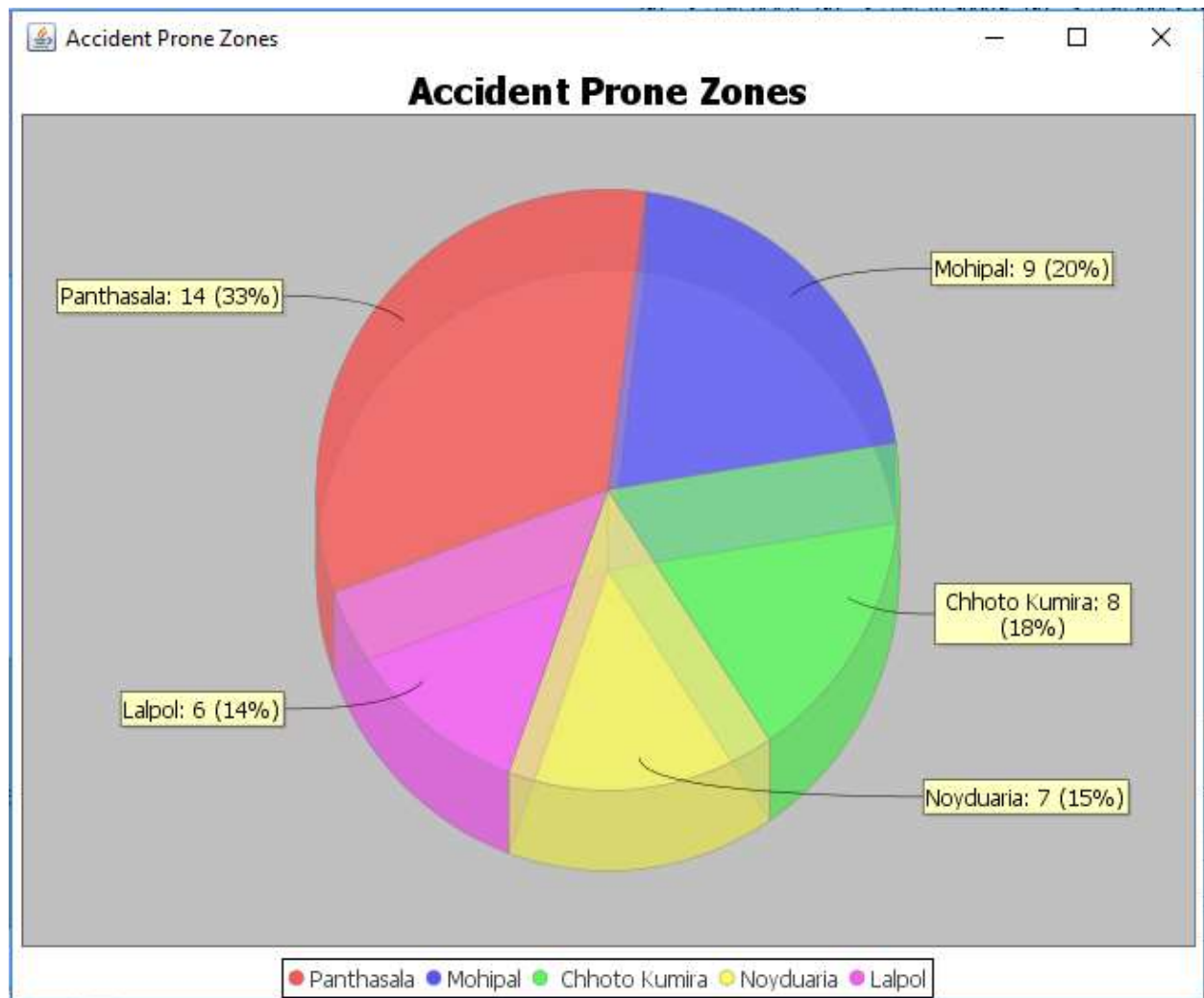


Fig.15. ACCIDENT PRONE ZONES

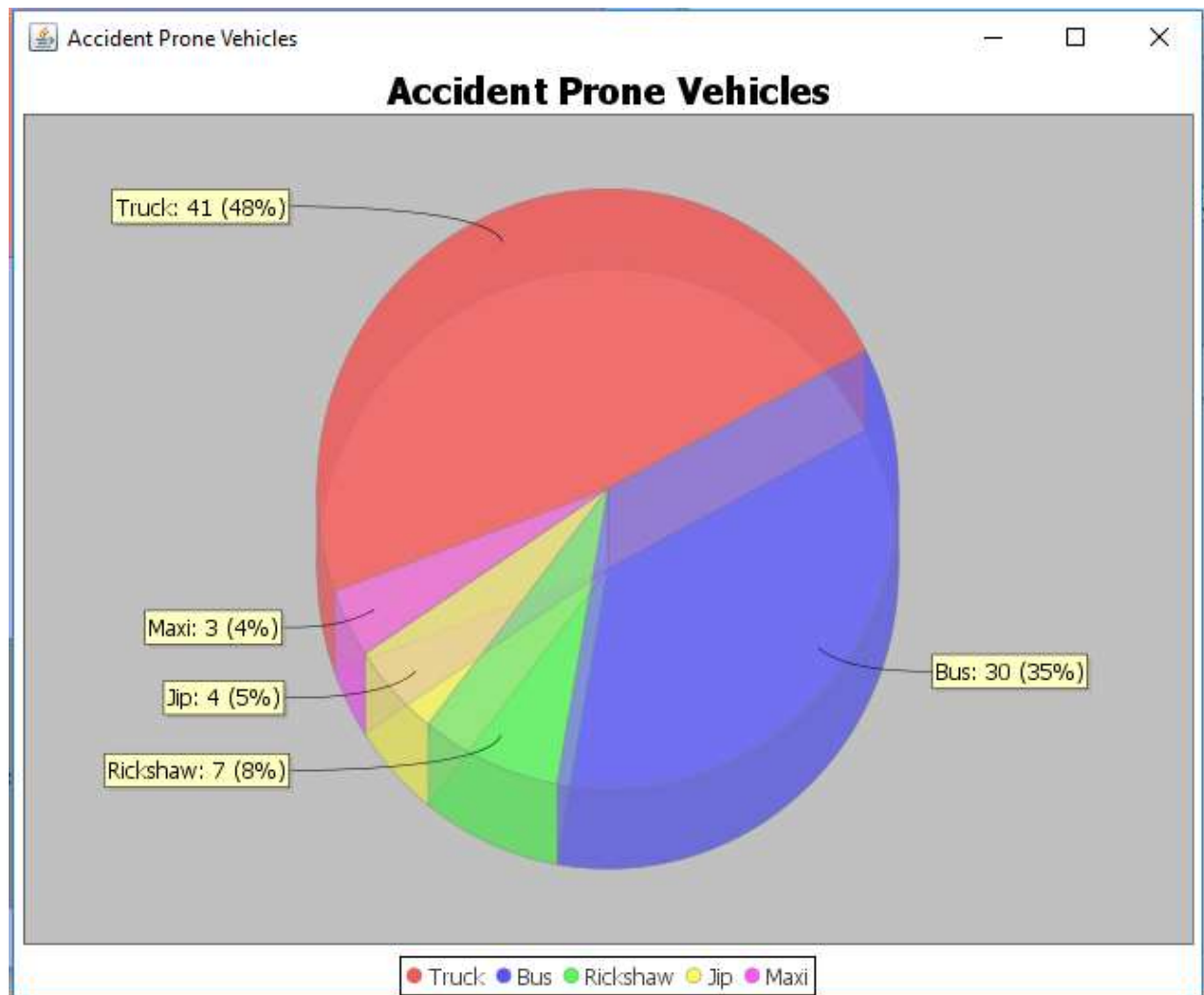


Fig.16. ACCIDENT PRONE VEHICLE

Prone Zone Name	Most Accidental Reason	Number Of Accident	Preventive Method
Panthasala	Face To Face Attack	39	
Mohipal	Face To Face Attack	26	
Chhoto Kumira	Face To Face Attack	24	
Noyduaria	Face To Face Attack	30	
Lalpol	Wrong Overtaking	14	

Back

Fig.17. SUGGESTED PREVENTIVE METHOD FOR PRONE ZONE

C. PLAGIARISM REPORT

